8-5-2020

Smarter Simulation Placement of Kilonova Light Curve Models for Computationally Inexpensive Surrogate Model Creation

Marko Ristic
mr3364@rit.edu

Smarter Simulation Placement of Kilonova Light Curve Models for Computationally Inexpensive Surrogate Model Creation

Marko Ristić

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Astrophysical Sciences & Technology

Astrophysical Sciences and Technology
College of Science
Rochester Institute of Technology
Rochester, NY
August 5, 2020

Approved by: _____

Andrew Robinson, Ph.D. Date
Director, Astrophysical Sciences and Technology

Approval Committee: Richard O'Shaughnessy, Joshua Faber, Yosef Zlochower

Astrophysical Sciences and Technology
College of Science
Rochester Institute of Technology
Rochester, NY

CERTIFICATE OF APPROVAL

M.S. DEGREE THESIS

The M.S. Degree Thesis of Marko Ristić
has been examined and approved by the thesis
committee as satisfactory for the thesis
requirement for the M.S. degree in Astrophysical
Sciences and Technology.

Dr. Richard O'Shaughnessy, Thesis Advisor

Dr. Joshua Faber, Committee Chair

Dr. Yosef Zlochower

Date

To my mother: Without you, none of this would have been possible.

Contents

Abstract

The first detected binary neutron star merger GW170817 allowed for the simultaneous detection of gravitational and electromagnetic waves which started the era of multi-messenger astrophysics. The existence of an electromagnetic counterpart to a compact object merger allowed for a significantly deeper analysis of the merger event and much tighter resultant constraints on existing physical models of neutron stars, nuclear physics, and the Universe itself.

Multi-messenger analysis requires sophisticated source modeling. For the foreseeable future, demanding computational resource costs will result in a sparse availability of state-of-the-art neutron star merger light curve simulations. Astrophysical inference can proceed using an alternate approach of creating computationally cheaper surrogate models based on the aforementioned state-of-the-art simulations.

The work presented here focuses on the creation and interpolation of a library of light curve simulations suitable for the generation of surrogate models capable of conveying useful astrophysical information. It addresses the necessity of switching from grid-based simulation placement to an error-maximization approach which identifies the least understood regions of parameter space. Interpolation is introduced as the connecting factor between the long-term goal of surrogate model creation and the new simulation placement mechanism. Finally, a discussion about the iterative process of simulation placement using interpolation outputs describes how each new simulation brings the library one step closer to serving as a surrogate model training set.

1 Introduction

Neutron stars are the densest known objects in the Universe, capable of holding twice the mass of the Sun in a region only tens of kilometers in diameter. They are created from Type II supernovae if the stellar core has enough degeneracy pressure to counter gravitational collapse. The conditions required for the densities found in neutron stars are irreplicable on Earth, making them an interesting subject matter for the study of physics in extreme environments. Although some neutron stars can be detected via their strong polar electromagnetic emission [1], systems in which two neutron stars are gravitationally interacting to form a neutron star binary convey important information which is not accessible by observing a single neutron star [2; 3; 4].

For nearly half a century, neutron star binaries have been known to exist in nature, stemming from the first detection of a binary pulsar system [5]. The existence of such binaries immediately invited the discussion of the unique physics which would occur in the extreme environments produced by the collision of these systems. Two gravitationally interacting objects not under the influence of an external force are expected to eventually collide as they lose energy through gravitational radiation. It is precisely this principle which has allowed the LIGO experiment to detect its first and many subsequent black hole mergers [6; 7]. Neutron stars are dense enough to emit detectable gravitational radiation, but not as dense as black holes such that no light can escape their gravitational pull. This crucial distinction allows for the possibility of electromagnetic radiation in addition to gravitational radiation during a neutron star merger [8]. With the famed detection of gravitational and electromagnetic signatures across the spectrum from the GW170817 binary neutron star merger, decades of theory had been confirmed and heralded the first ever "laboratory experiment" involving neutron star binaries [9; 10; 11; 12; 13; 14].

The combination of gravitational and electromagnetic wave detections allows for much more informative conclusions about the underlying physics of neutron stars and the Universe. The emission of electromagnetic signatures from the merger allows for precise localization of the merger event to a nearby galaxy. Comparison of the host galaxy's redshift to the true distance to the source recovered from the gravitational wave data allow for a direct measurement of the Hubble constant independent of the cosmological distance ladder or cosmic microwave background methods [15]. In addition, observed electromagnetic emission exhibits different patterns based on the elemental composition of the ejected material. This allows for better understanding of the production of elements heavier than iron through the rapid neutron-capture process [16; 17]. A similar approach is applied to constraining feasible candidates for the nuclear equation of state [18; 19].

In order for the aforementioned constraints to be made, there must exist a comparison of observations to an accurate and representative theoretical model of the electromagnetic emission. The identification of such a model requires information regarding the pre-merger binary system and the post-merger remnant and environment. This information is carried in the observed electromagnetic and gravitational radiation. However,

a given pattern of emission is not exclusive to the combination of parameters which generated it using a specific model. As such, the inferred parameters for a given merger event can at best be represented by a distribution of possible values which peaks at the most likely estimate. The range of the distribution of possible values depends on the uncertainty at some level of confidence. The accuracy of this parameter inference process is essential as it is responsible for tighter constraints on the physics of neutron stars, nuclear matter, and the Universe.

Representative theoretical models are hard to solve analytically and require the incorporation of numerical simulations. Due to the complexity of theoretical models involving binary neutron star mergers, their electromagnetic emission, and the underlying microphysics calculations which require vast computational resources, state-of-the-art simulations are expensive and available in sparse quantities. This makes the task of parameter inference more difficult, as there are fewer models to which observations can be compared. In order to achieve the maximum level of accuracy and constraint from the theoretical models, there is a clear need for an alternate approach to producing simulation-quality outputs at a fraction of the computational cost. One possibility would be to use all existing simulations as a training set and interpolate between them in order to create cheaper surrogate models. However, successful and accurate interpolation requires a large enough training set to allow for generalization across the parameter space.

The purpose of this thesis is to address the disparity between the sparsity of available binary neutron star merger light curve simulations and the large training set requirement of accurate interpolation outputs. The first section will discuss the physics background surrounding the post-merger environment and kilonovae. The second section will explain the details of the simulations used and the initial grid of simulations on which the surrogate model interpolators were trained. The third section will cover the intricacies of the interpolators used and distinguish their differences. The fourth section will detail the implementation of the work so far. The fifth and final section will discuss planned future work, specifically upgrades to existing tools and methods.

2 Kilonovae

Neutron stars mergers are predicted to expel significant amounts of ejecta in all directions through highly energetic processes born from the relativistic and compact nature of the stars [20; 21]. As matter is ejected, it forms a dense and highly energetic environment which serves as the site of the rapid neutron-capture process, or r-process, in which atomic nuclei gather neutrons at a rate faster than the timescale of radioactive decay. As a result, the ejecta becomes very neutron rich, and the electron fraction, the ratio of proton to the sum of proton and neutron number densities,

$$Y_e \equiv \frac{n_p}{n_p + n_n}, \tag{1}$$

quickly drops in value. If released on timescales comparable to the merger timescale, this neutron-rich ejecta can be categorized as dynamical ejecta. Dynamical ejecta is released via two main channels: the first occurs at the contact interface between the two stars where material is squeezed out and spreads isotropically as the remnant pulsates while the second results from tidal forces spreading ejecta in the orbital plane and creating a spiral arm structure [22]. The planar ejecta forms an accretion disk around the remnant which releases neutron-rich wind ejecta, the counterpart to dynamical ejecta, as it cools.

The amount of ejecta, and thus electromagnetic radiation, released depends largely on the behavior of the post-merger remnant and its lifetime. Depending on its mass and rotational velocity, the remnant will take one of four forms as explained in [21]. To summarize, if the remnant is substantially heavy and without the necessary rotational velocity to combat gravitational collapse, it will immediately collapse into a black hole with little to no electromagnetic emission. If there is substantial rotational velocity to prevent collapse for $\mathcal{O}(10-100)$ ms, a hypermassive neutron star remnant is formed. Similarly, if the aforementioned remnant survives for considerably longer than $\mathcal{O}(100)$ ms prior to collapse, it forms a supramassive neutron star remnant. Finally, if the remnant is not so massive that its internal pressure cannot support itself against the effects of strong-field gravity (for non-rotating neutron stars what is known as the Tolman-Oppenheimer-Volkoff limit, [23]), it can survive as a regular neutron star without ever collapsing into a black hole as the other remnants do.

Over time, the ejecta expand away from the merger remnant at a fraction of the speed of light and become less optically thick. As the remnant cools through neutrino release, the disk is heated by the bombardment of the emitted neutrinos. The disk in turn cools by re-emitting neutrinos as wind ejecta and heating the surrounding far-flung ejecta, composed of neutron-rich radioactive elements, but also deposits energy back onto the remnant through accretion [24; 25]. The surrounding neutron-rich ejecta then cools through radioactive decay and creates the kilonova which releases the distinct electromagnetic emission indicative of neutron star mergers. It is this feedback between the merger remnant and the accretion disk that causes the lifetime of the merger remnant to directly influence the kilonova emission.

2.1 Introductory Toy Model

The model for the ejecta outflows in the post-merger environment follows the assumptions of [26] and [27] and subsequent publications [21; 28]. The physical assumptions are i) a homologous, spherically symmetric expansion of the ejecta, ii) a radiation pressure dominated environment, and iii) a central engine providing some form of input energy. A simple order-of-magnitude calculation for the peak brightness time is initially presented, following the approach from Section 3 of [21], with a detailed follow-up discussion of the physical processes at hand.

The recipe in Section 3 of [21] establishes a simple physical setup for the ejecta outflows and the following derivation follows that approach exactly. First, merger ejecta of mass M is assumed to expand spherically outward at constant average velocity v, such

that after time t, it has reached a mean distance of $R = vt$. The spherical symmetry argument follows from the fact that by the time the optical depth of the ejecta is thin enough such that radiation can escape, the overall shape of the ejecta will have smoothed out any initial asymmetries. At this time, the optical depth τ can be approximated by

$$\tau \simeq \rho\kappa R = \frac{3M\kappa}{4\pi R^2} \tag{2}$$

where ρ is the mean density $\rho = \frac{M}{\frac{4}{3}\pi R^3}$ and κ is the opacity, as in Equation (5) of [21]. Translating these quantities to a physical scaling relation, the optical depth can be written as

$$\tau \simeq 70 \left(\frac{M}{10^{-2}M_\odot}\right)\left(\frac{\kappa}{1\ \mathrm{cm^2 g^{-1}}}\right)\left(\frac{v}{0.1c}\right)^{-2}\left(\frac{t}{1\ \mathrm{day}}\right)^{-2} \tag{3}$$

from which it is apparent that for the majority of post-merger scenarios, the optical depth will initially be very thick. To determine the approximate diffusion timescale, the ejecta is assumed to traverse the mean distance R at the speed of light c such that

$$t_{\mathrm{diff}} \simeq \frac{R}{c}\tau = \frac{3M\kappa}{4\pi cR} = \frac{3M\kappa}{4\pi cvt} \tag{4}$$

with τ naturally present as a scaling factor which prolongs the diffusion timescale for thicker optical depths. Once the diffusion timescale is comparable to the expansion timescale, i.e. once $t_{\mathrm{diff}} = t$, then radiation can begin escaping out from the ejecta. With this, Equation 4 can be re-written as

$$t^2 = \frac{3M\kappa}{4\pi cv} \tag{5}$$

for convenience. Realizing that the intensity of the radiation will be brightest when it finally escapes the ejecta (dimmer prior because it could not escape and dimmer after escape because energy leaves the system), Equation 5 can be re-written to represent the light curve peak time as

$$t_{\mathrm{peak}} \equiv \left(\frac{3M\kappa}{4\pi\beta vc}\right)^{\frac{1}{2}}, \tag{6}$$

where β is a constant dependent on the ejecta density profile for a given scenario.

These equations, obtained from Equations 5–7 in [21], offer convenient characteristic values for the optical depth and peak time dependent on ejecta parameters. It is important to remember that, up to this point, this is a toy model largely based on simple assumptions with little consideration for the underlying physical processes.

Equation 6 can be re-written as a scaling relation of the form

$$t_{\mathrm{peak\ [21]}} \approx 1.6\ \mathrm{d}\left(\frac{M}{10^{-2}M_\odot}\right)^{\frac{1}{2}}\left(\frac{v}{0.1c}\right)^{-\frac{1}{2}}\left(\frac{\kappa}{1\ \mathrm{cm^2 g^{-1}}}\right)^{\frac{1}{2}}, \tag{7}$$

which allows comparison to a similar model presented in [29] suggesting that

$$t_{\text{peak [29]}} \approx 4.9 \text{ d} \left(\frac{M}{10^{-2} M_{\odot}}\right)^{\frac{1}{2}} \left(\frac{v}{0.1c}\right)^{-\frac{1}{2}} \left(\frac{\kappa}{10 \text{ cm}^2\text{g}^{-1}}\right)^{\frac{1}{2}}. \tag{8}$$

The scaling relation from [30], which describes single-dimensional variants of the models discussed in this work, varies slightly from the previous two with the peak time approximated by

$$t_{\text{peak [30]}} \approx 1.0 \text{ d} \left(\frac{M}{10^{-2} M_{\odot}}\right)^{0.318} \left(\frac{v}{0.1c}\right)^{-0.60} \left(\frac{\kappa}{10 \text{ cm}^2\text{g}^{-1}}\right)^{0.35}, \tag{9}$$

though it yields a much earlier peak time prediction due to its inclusion of the thermal contribution to the light curve at early times. Section 3.1 of [30] describes the nuances of the presented scaling relation. It is important to note that Equation 9 pertains to the semi-analytic 1-dimensional models; the complexities involved with the two-component models with non-spherical morphology discussed in this work do not allow for as simple a representation of the peak time as shown in Equations 7 – 9. Table 2 of [31] shows representative peak times for two-component models with a variety of sampled morphologies.

2.2 More Realistic Model Description

A more advanced approach to the physics governing the post-merger system, still considering the assumptions from the previous section, starts with the first law of thermodynamics,

$$dU = \partial Q - P\partial V \equiv dU + P\partial V = \partial Q, \tag{10}$$

where U is the internal energy of the system, P is the pressure, V is the volume, and Q is the heat added to the system. The heat added to the volume comes from the third assumption of the toy model which states that there is a central engine providing energy to the system as well as other sources such as radioactive decay heating [32]. The system, however, is simultaneously losing energy via emitted photons generated by way of radioactive decay. Thus, Equation 10 can be re-written as

$$\dot{E} + P\dot{V} = \epsilon_{\text{in}} - \frac{\partial L}{\partial m}, \tag{11}$$

with $\dot{E}$ representing the internal energy change, ϵ_{in} representing the input energy from the central engine, and $\frac{\partial L}{\partial m}$ representing the radiation emitted by radioactive isotope decay.

To better understand the behavior of this radiation driven energy loss, the diffusion equation describing the flow of photons under the effect of radiation pressure is

considered, with Equation (9.31) from [33] giving

$$\frac{dP_{\text{rad}}}{dr} = -\frac{\kappa\rho}{c}F_{\text{rad}} = -\frac{1}{\lambda c}F_{\text{rad}} \tag{12}$$

where P_{rad} is the radiation pressure, κ is the average opacity, ρ is the density, and c is the speed of light. For convenience, $\lambda = \frac{1}{\kappa\rho}$ is used in the second equation, where λ is the mean free path for some photon traveling through a medium with average opacity κ and density ρ.

For a radiation dominated medium, the energy energy density is $u = aT^4$ and conversely the pressure is

$$P_{\text{rad}} = \frac{u}{3} = \frac{aT^4}{3} \tag{13}$$

which, when substituted into Equation 12, with a re-definition of radiative flux in terms of luminosity and some light re-arranging, yields

$$\frac{L}{4\pi r^2} = -\lambda c a \frac{\partial\left(T^4\right)}{\partial r} \tag{14}$$

which precisely matches Equation (2) of [27] and Equation (2) of [28]. Equation 14 is particularly important as it, in conjunction with Equation 11 allows for the solution of the partial differential equations which yield general analytical light curve models. While the exploration of the solutions for these differential equations is beyond the scope of this work, [27] and [28] go into extensive detail regarding the nature of the solutions. Further discussion on radioactive decay heating rates is explored in [34].

2.3 Observing Kilonovae

Figure 1 shows a cartoon depiction of the post-merger components which are most relevant to the discussion in this thesis. Note that this is a purposely over-simplified schematic and is not meant to represent all aspects of the post-merger environment (Figure 1 of [35] is more comprehensive). Each component of the ejecta, dynamical in purple and wind in green, has its own mass and velocity parameters.

An example of simulated kilonova electromagnetic emission is shown in Figures 2 and 3. These figures serve to establish the general profile and behavior of these light curves in each bandwidth of interest. The first figure shows the log bolometric luminosity, obtained when electromagnetic emission is integrated over all relevant wavelengths, as a function of time. The second figure shows the log luminosity in each of the wavelength bands, noted by the different letters representing filter colors, as a function of time. Table 1 indicates the bandwidths and mean wavelengths for each filter.

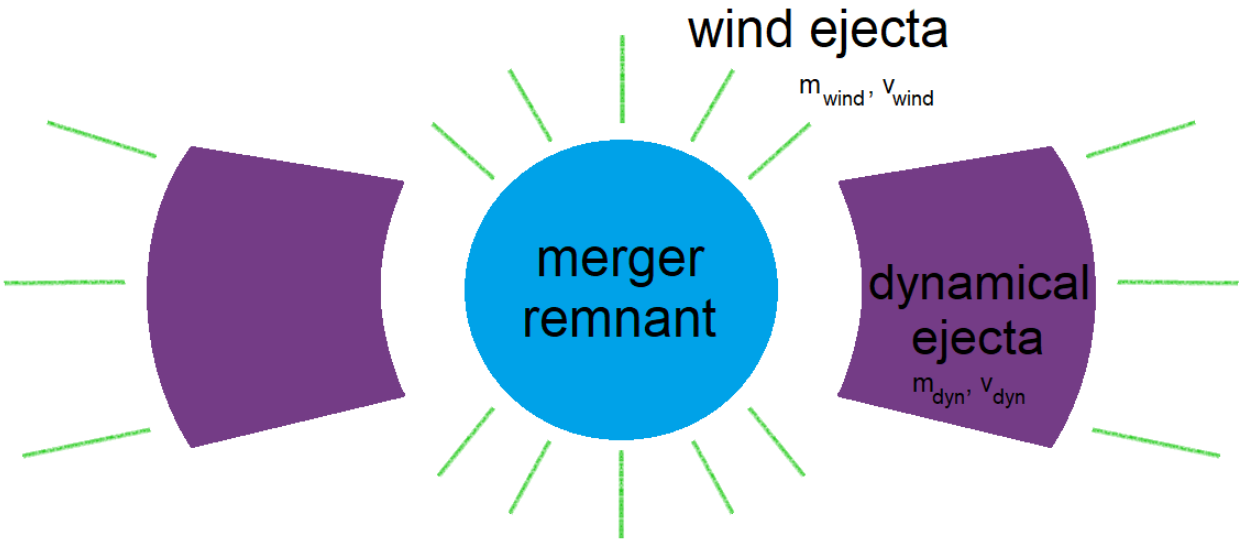

Figure 1: Cartoon image showing the post-merger environment with the two ejecta components and their respective parameters identified. The dynamical ejecta in purple is composed of an accretion disk formed by the removal of neutron star matter by tidal forces as well as any material squeezed out from the progenitor interface during the merger event. The wind ejecta in green accounts for all other sources of ejecta which are primarily driven by neutrino outflows from the remnant (prior to collapse) and accretion disk.

Filter	λ_{mean} (nm)	Width (nm)
g	478.1	87.5
r	621.2	80
i	754.2	77
z	871.2	67
y	1003.6	94
J	1251.9	198.1
H	1637.9	276.1
K	2175.5	223.5
S	4393.6	823.1

Table 1: Mean wavelengths and bandwidths for the nine broadband filters in which light curves are produced. Bands 'g' through 'y' are based on LSST filters [36], 'J' through 'K' are based on 2MASS filters [37], and 'S' is based on the Spitzer 4.5μm wavelength [38].

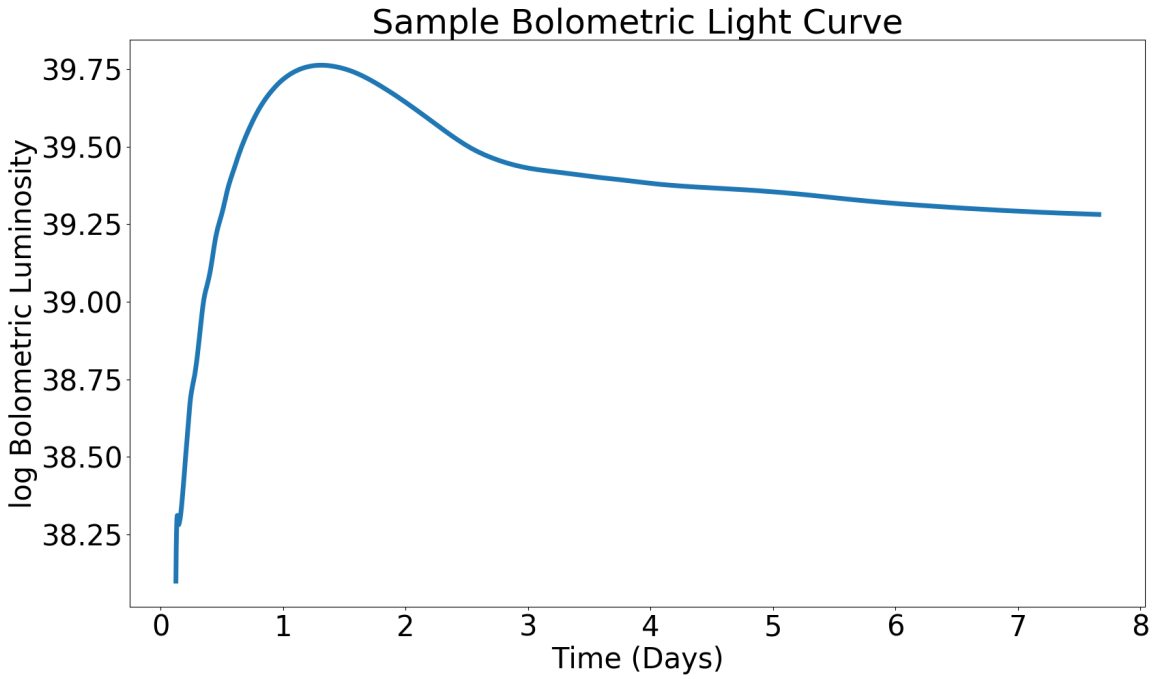

Figure 2: Example simulation of the bolometric (integrated across all wavelengths) luminosity as a function of time with dynamical mass = 0.01 $M_\odot$, dynamical velocity = $0.3c$, wind mass = 0.01 $M_\odot$ and wind velocity = $0.3c$. The emission rises rapidly within the first day as a result of intense energy dissipation into the surrounding material and peaks around 1–2 days post-merger. The long, slowly-decaying tail starting at 2 days is composed primarily of redder wavelength emission ($\sim$ 650 nm and longer) which radiates away on a longer timescale than emission at shorter, bluer wavelengths.

3 Simulated Light Curve Models

When combined, the physics recipes mentioned in the previous section allow for the creation of light curve models of varying complexity. While different approaches will consider their own respective assumptions, the general scenario is that of [27] in which homologous expansion of merger ejecta occurs in a radiation pressure dominated environment. Typically, the different ejecta are split up into components as in Figure 1 due to differences in formation mechanisms as well as evolutionary timescales. Each of these components has separate parameters which describe them, namely the ejecta mass, velocity, nuclear composition, and large-scale morphology, or shape. Numerous groups have developed physical models using these ejecta parameters to create simulation outputs of kilonova light curves [30; 39; 19; 11]. One such physical model presented in [30], created by a group at the Los Alamos National Laboratory (LANL), generated the outputs in Figures 2 and 3. The LANL approach utilizes radiative transfer instead of hydrodynamics with the notable distinction that the opacities are specifically calculated

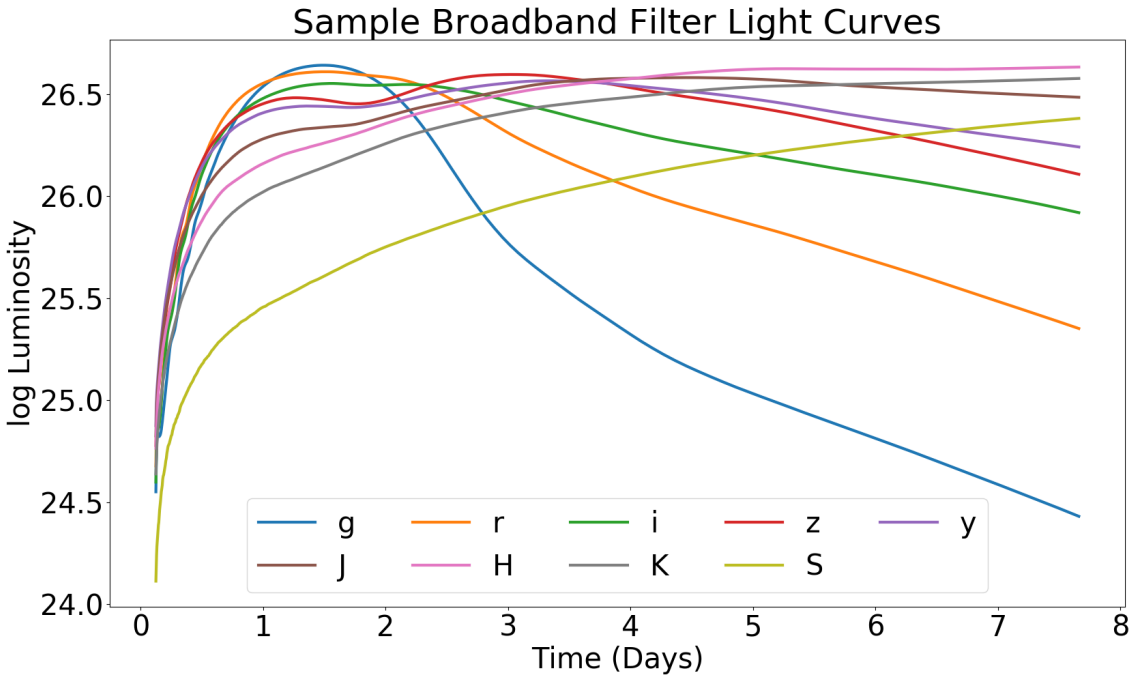

Figure 3: Example simulation outputs of the light curves as they would be observed in the respective broadband filters whose mean wavelengths and bandwidths are described in Table 1. The bands go from blue visible light (480 nm 'g' filter) to the mid-infrared (4.5 μm Spitzer channel 2 'S' filter). Note the different peaking behavior of the curves with bluer wavelengths peaking within days of post-merger and redder light curves peaking several days later.

rather than assumed to be a convenient constant value [40; 41].

3.1 Simulation Model Description

The simulations discussed throughout were generated using the SuperNu time-dependent radiation transfer code [42]. Extensive discussion regarding the physics and code methodology can be found in Section 2 of [30], but for the purposes of this thesis, a simplified summary of Section 2.1.1 of the same paper will suffice. The model is based around a symmetrically shaped ideal fluid expanding in vacuum described by the Euler equations of ideal hydrodynamics. The assumption of a radiation-dominated polytropic equation of state, along with other mathematical simplifications, allows for an analytic representation of the ejected mass and average velocity as a function of initial central density, initial time, and the velocity of the expansion front (Equations 11 and 12 in [30]). These equations yield two parameters which depend closely on the post-merger environment and predict long timescale evolution based on a simple model. When combined with Monte Carlo-based radiative transfer and a specific elemental composition, the code

outputs light curves such as the ones in Figures 2 and 3.

Due to the difference in formation mechanisms of dynamical and wind ejecta [21], a multi-component approach is necessary for accurate modeling. Each type of ejecta, dynamical and wind, is modeled by a separate component with a specified morphology, elemental composition, ejecta mass and ejecta velocity. The radiation emitted by each component is averaged with weighting in spatial cells occupied by both components, but otherwise the two are considered separately. The end light curve product is a representation of the radiation contained in each angular bin at every time step for either some wavelength or, in the bolometric case, the integration over all wavelengths.

Each output contains luminosity and magnitude data in the 'grizyJHKS' bands at a couple hundred time steps spanning from hours to weeks post-merger. The luminosity data includes an additional bolometric "band" component which integrates the emission over all available wavelength bands. The output also contains spectra which are composed of 1024 wavelength bins per time step. The time steps are unevenly sampled, notably being more dense in the earlier part of the light curve to better capture the detailed physics occurring shortly after the merger event. For each time step, 54 viewing angles, spaced equally in solid angle (and thus also in $\cos\theta$), are considered, with an independent light curve representing each viewing angle.

The degeneracy of different parameter combinations yielding very similar light curve outputs makes for difficult prediction of choosing representative parameter combinations for an initial collection of light curves. The benefit of sampling over a grid is that it allows for a controlled approach which may highlight patterns or features based on equally-spaced (ejecta mass), extreme (ejecta velocity), or functionally different (morphology and elemental composition) parameter choices. This approach was taken by the LANL group in creating their first group of kilonova light curves on a grid.

3.2 Simulations

As shown in Figure 3, the different wavelength filters show peak emission at different timescales. From the broadband filter curves, it is apparent that the dissipation of energy is much quicker for the higher energy, shorter wavelength photons than their redder, longer wavelength counterparts. Some form of this behavior will manifest in every set of light curves produced, but an added assumption is that the overall shapes and profiles of the light curves will differ based on the ejecta properties.

With this assumption in mind, the goal is to determine some deterministic relationship between the properties of the post-merger ejecta and the observed light curves. Previous work has shown that dynamical and wind ejecta properties, namely mass and velocity, have a direct impact on the kilonova light curve profiles [13; 43; 44].

A grid of simulations was generated over different combinations of physical parameters in a two-component ejecta model. These parameters are the wind ejecta morphology, wind elemental composition, dynamical ejecta mass and velocity, and wind ejecta mass and velocity. The wind ejecta morphology was either spherical or peanut-shaped, while the wind ejecta composition had two possible models with a higher $Y_e = 0.37$ and

lower $Y_e = 0.27$ electron fraction, respectively; the dynamical ejecta composition and morphology were permanently fixed to be $Y_e = 0.04$ and torus-shaped, respectively. The masses were log-spaced equally between 10^{-3} $M_\odot$ and 10^{-1} $M_\odot$ and the velocities were representative of low or high ejecta speeds at 0.05c and 0.3c, respectively. Table 2 shows all the parameter combinations considered in the creation of the grid.

At this point, it is necessary to note an extremely important detail regarding the consideration of the initial grid for the remainder of this thesis. While the full grid consisted of 144 simulations acquired through every combination of the parameters in Table 2, only a subset of this whole grid was actually used in the work presented here. The wind ejecta morphology and elemental composition were fixed to the peanut-shaped and lower $Y_e = 0.27$ composition, respectively; this simplified the problem to one of only regression across the ejecta mass and velocity parameters whilst removing the complications of classification involving morphologies and compositions. Thus, from this point onward, all comments made are regarding this specified subset containing only 36 simulations generated using the bolded parameters in Table 2. For simplicity, this subset will be referred to as the "initial grid" from this point forward. Any non-specific reference to "the grid" can be understood as referring to the full grid of 144 simulations.

3.3 Simulation Results

Figure 4 shows the bolometric light curves for the mass and velocity parameter combinations of the initial grid. The curves separate into groups evident several days post-merger which seem to separate into high-, middle-, and low-luminosity categories. These categories are roughly represented by late-time (past 7 days) luminosities of $\log(L_{\rm bol}) > 39$, $38 < \log(L_{\rm bol}) \leq 39$, and $\log(L_{\rm bol}) \geq 38$, respectively. Zooming in only up to the first day post-merger in Figure 5 indicates the presence of a different set of groupings at early times as well. These groupings point to patterns which emerge based on the input parameters – their presence in these curves stems from the equal log-mass spacing and binary velocity selection. A more in-depth discussion on light curve features emerging from grid-based parameter choices continues in Appendix A.

This initial recognition of grouping patterns emerging as a result of input parameters was indicative of the necessity of a different approach to future parameter choices. Two things became apparent, the first being that a more varied set of light curves would be achieved if the discrete grid-like parameter selection was replaced with random sampling of parameter combinations within a range of physically realistic values. The second was that, due to their grouped nature, grid-based simulations were not maximizing the amount of information obtained about the parameter space with each new simulation. Ideally, a new approach for simulation placement would consider both these issues and choose parameters which are continuous and explore as much of the parameter space as possible.

Parameter	Grid Values
Dynamical Composition	$\boldsymbol{Y_e = 0.04}$
Dynamical Morphology	**Torus**
Wind Morphology	Spherical, **Peanut**
Wind Composition	$Y_e = 0.37$, $\boldsymbol{Y_e = 0.27}$
Dynamical Mass	**0.001, 0.01, 0.1** [$M_\odot$]
Dynamical Velocity	**0.05, 0.3** [$\boldsymbol{c}$]
Wind Mass	**0.001, 0.01, 0.1** [$M_\odot$]
Wind Velocity	**0.05, 0.3** [$\boldsymbol{c}$]

Table 2: Parameters used in the creation of the first kilonova light curve simulation grid. The product of all the parameter combinations yields a full grid of 144 simulations. The initial grid discussed in the text refers to the subset of 36 simulations which have a fixed peanut wind morphology and the lower $Y_e = 0.27$ wind composition. The parameters considered in the creation of the initial grid and the simulations discussed in this work are bolded for clarity.

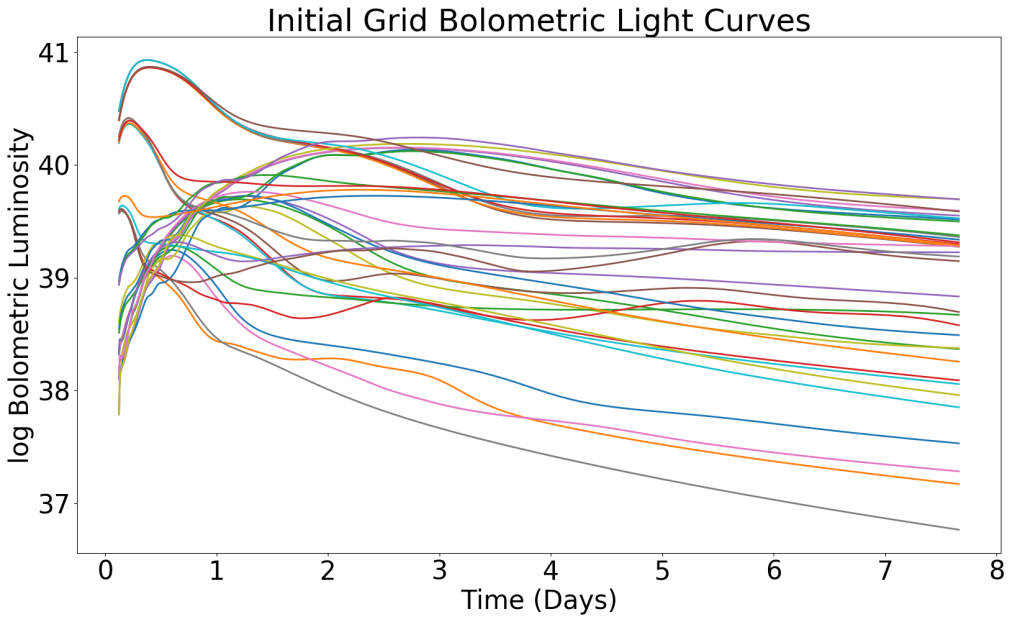

Figure 4: Light curves of the initial grid prior to addition of placed simulations. The most prominent groupings in early-time are due to the wind ejecta parameters, whereas the dynamical ejecta parameters dictate the late-time behavior (see Appendix A).

4 Interpolation

Interpolation was chosen as the method which would address the short-term problem of requiring smart placement of new simulations as well as the long-term goal of providing

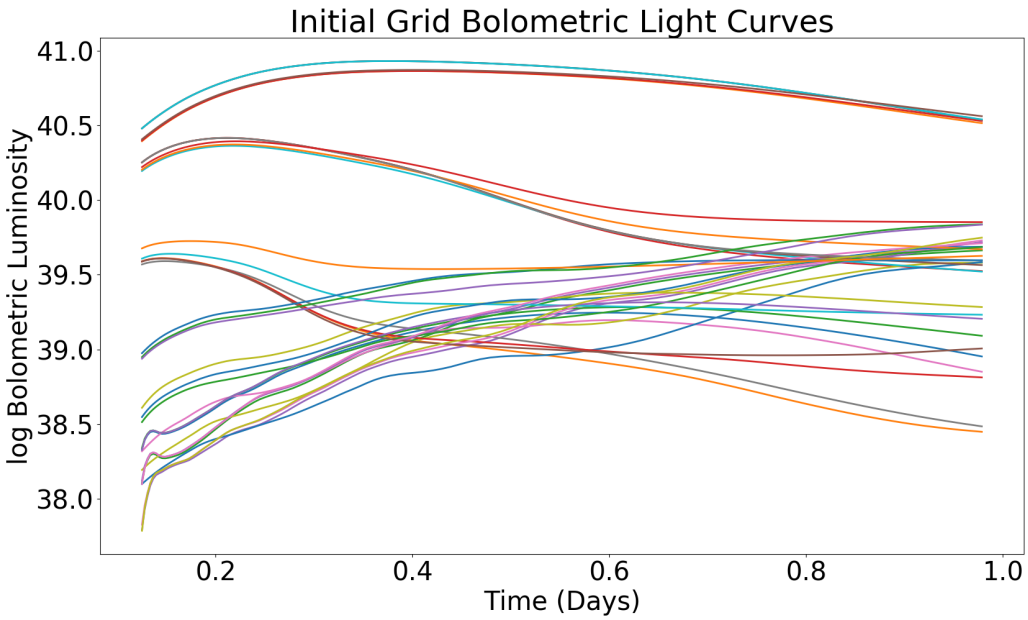

Figure 5: Zoom-in of Figure 4 light curves only up to 1 day post-merger with overlapping curves signifying groupings due to similar grid-based parameter selection.

novel astrophysical results via estimation of light curves between existing simulations to some degree of reasonable accuracy. Interpolation could help with simulation placement by training on the current state of the simulation library and evaluating which parameter combinations are unable to be interpolated to the desired level of uncertainty. Eventually, with a large enough training library, it would allow for the creation of simulation-like outputs using a fraction of the computational power with the trade-off of having uncertainty determination unlike the simulation outputs. In order to minimize the degree to which this surrogate model error affected the results, there first needed to exist a metric which could quantify this error. A suite of interpolators would then be used to minimize this error metric using four different algorithms: the Gaussian process, random forest, bagging regressor, and neural network. The purpose of using different algorithms is to ensure that the interpolations are general (rather than working by chance for a specific algorithm) and to ensure fidelity, both of interpolation consistency between interpolators and for future studies when the data volume will presumably be large. Finally, the application of the suite of interpolators via an iterative process, discussed in Section 5, will enable the generation of a suitable training library for astrophysical surrogate model creation.

4.1 Performance Metric

First, it was necessary to establish a metric which would gauge how well any given interpolator could evaluate a certain set of inputs given some training period. Obviously, the interpolators are expected to learn from the training set; this means that if supplied with a collection of parameters matching those in the training set, it should produce an incredibly similar output to the training data. Since the interpolators do not know the true function mapping the inputs to the outputs, there will be some degree of error in each evaluation. A simple representation of this error is the mean squared error

$$\text{error} = \frac{1}{n}\sum_{j=1}^{n}(y_j - \log(L_{\text{bol}})_j)^2, \tag{15}$$

with n number of samples, y_j representing the interpolator's approximation given some inputs, and $\log(L_{\text{bol}})_j$ being the simulation output for the same inputs. It is obvious from Equation 15 that as the interpolator's approximation approaches the true simulation output value, the error slowly decreases to zero. It is worth noting that, for simplicity, the simulation outputs were assumed to have no error throughout this work.

Each interpolator uses a different approach to minimizing the error between the training data and the interpolator outputs. Of the four interpolators discussed below, only the random forest and bagging regressor use the mean squared error from Equation 15 for optimization during training. The Gaussian process uses an L-BGFS-B optimizer [45] to constrain the distribution of possible functions and minimizes the standard deviation of this distribution which it also reports as an uncertainty on the output with each evaluation. The neural network minimizes the reduced chi-squared goodness-of-fit test statistic with an adjustment that favors more precise fits near the peak of the light curve while also reporting an error on each evaluation (see Section 4.2.4 below).

While the above discussion highlights how each individual interpolator minimizes the error between the training data and the interpolator's predictions, the more relevant information lies in how these errors affect the placement of new simulations. The connection between these two concepts was made by realizing that the training library was insufficiently populated in regions where the interpolators could not make accurate, or any, predictions and thus yielded large prediction errors. As such, the metric used to gauge the training library's coverage of the parameter space was the maximization of the interpolator prediction error (which was, ironically, the exact opposite of what the interpolators attempt to accomplish during training). By choosing parameters which corresponded to the largest interpolator prediction errors, new simulations were placed in regions of phase space where the interpolators performed least reliably. This placement scheme achieved the previously set goals of populating sparsely-sampled regions of parameter space while simultaneously maximizing the information provided by each placed simulation.

4.2 Suite of Interpolators

A suite of interpolators was considered for investigation of the initial grid's capabilities as a training set. While most of the interpolation tools considered in this work were accessed from the scikit-learn library of machine learning algorithms for Python [46], the neural network was constructed separately without the use of a pre-built model using the PyTorch framework [47]. The nature of the Gaussian process (GP), discussed below, made it a logical candidate for the interpolator which reported the error of any prediction. The other two scikit-learn interpolators, the random forest (RF) and bagging regressor (BR), were implemented as a consistency check to ensure that the GP interpolation was not erroneous due to interpolator-specific sensitivities. The neural network was not used for the results presented in this work, however it will be the main interpolation and placement mechanism in future work; as such, it is discussed for fidelity and ease of discussion in Section 7.

To establish notation, all of the interpolators mentioned below function as a map from a d-dimensional input vector of physical parameters $\mathbf{p} \in \mathbf{R}^d$ to a vector of log bolometric luminosity values $\log(\mathbf{L_{bol}}) \in \mathbf{R}$. The training targets $\log(\mathbf{L_{bol}})_{\text{true}}$ are first pre-processed to have zero mean and unit variance such that

$$\log(\mathbf{L_{bol}})_{\text{train}} = \frac{\log(\mathbf{L_{bol}})_{\text{true}} - \log(\bar{\mathbf{L}}_{\mathbf{bol}})_{\text{true}}}{\sigma_{\log(\mathbf{L_{bol}})_{\text{true}}}}. \tag{16}$$

The structure of the inputs and outputs varies depending on what training parameters are being used, but generally the inputs are a two-dimensional array of shape $(N \times d)$ and the outputs are an array of shape $(N \times t)$, where d are the physical training parameters and t is the number of time steps.

Simply put, the interpolators were provided dynamical mass, dynamical velocity, wind mass, and wind velocity values as training inputs $\mathbf{p}$. The outputs were full log bolometric luminosity light curves evaluated at the fixed time points provided by the simulations except for cases where time was used as a training variable (see Section 4.2.4). Each combination of input parameters $\mathbf{p}$ mapped to exactly one log bolometric luminosity light curve $\log(\mathbf{L_{bol}})$ to ensure no degenerate behavior. The error provided by the Gaussian process was a single scalar value representative of the accuracy of the entire light curve. The approach of using fixed time points is not optimal as different regions of the light curve may be better fit than others, but it is considerably more efficient than any other alternative considered in this work.

4.2.1 Gaussian Process

The following is a simplified summary of the information presented in Chapter 1 of [48] which is pertinent to this work. A Gaussian process (GP) is a distribution of all possible functions which may fit to the training data. Assuming no prior knowledge about the data, as in the left side of Figure 6, the prior distribution of functions which may represent the data is uniform and infinite. It is safe to assume, then, that the

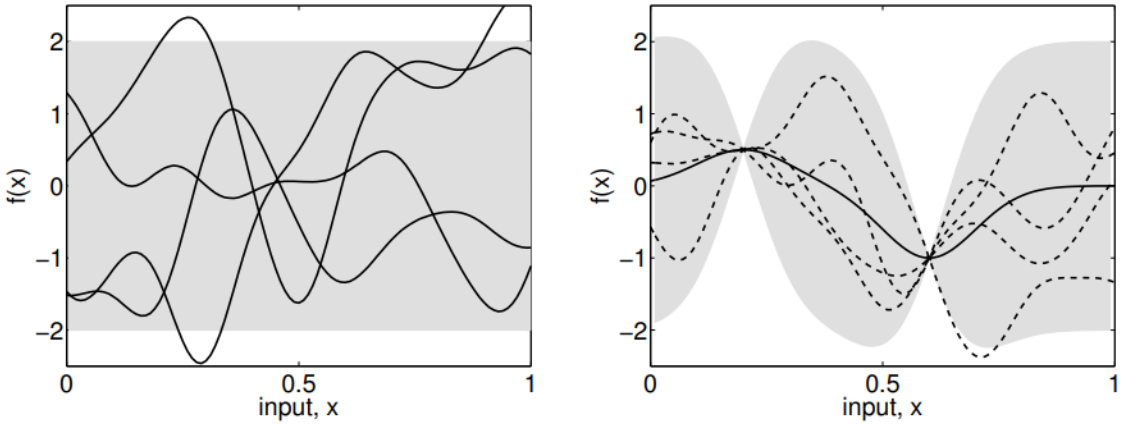

Figure 1.1: Panel (a) shows four samples drawn from the prior distribution. Panel (b) shows the situation after two datapoints have been observed. The mean prediction is shown as the solid line and four samples from the posterior are shown as dashed lines. In both plots the shaded region denotes twice the standard deviation at each input value x.

Figure 6: Figure 1.1 from [48], with original caption, describing prior (before training) and posterior (after training) distributions of functions which fit to the training data.

distribution of representative functions has a mean of 0 and a uniform value for twice the standard deviation for any given input. With the introduction of training data, as in the right side of Figure 6, the distribution of functions is naturally constrained and limited to those functions which must include the training data values at the respective inputs. The behavior of the functions in the regions where no data is present is left unconstrained and, as shown by the shaded lines, could take on any form so long as it does not violate the constraints set by the training data. The output of the GP is the mean function of all possible functions, represented by the solid line.

The innate assumption that there exists structure between certain inputs and outputs is essential for GPs, or any interpolators for that matter, to successfully make predictive models. There is an underlying expectation that very small changes in the input parameters will result in very small changes in the outputs. For GPs, this expectation manifests in the form of the covariance function, or kernel, which dictates the relationship between inputs by defining a distance metric. As the distance between two inputs increases, their covariance decreases, and the likelihood of the output changing in a less predictable manner increases. This is evident in the right plot of Figure 6 where the uncertainty of the mean function increases with the distance from the training points.

Given the infinite possibilities for functions which may fit to the training data, the GP attempts to minimize the uncertainty by finding the optimal hyperparameters in a given kernel. For the GP used in this work, the kernel was composed of a white

noise kernel added to a squared-exponential kernel which itself was scaled by a constant kernel. This manifested as

$$k(x_i, x_j) = \text{noise\_level}\big|_{x_i=x_j} + C \times \exp(-\frac{d(x_i, x_j)^2}{2l^2}), \tag{17}$$

where $k(x_i, x_j)$ is the kernel evaluated for two inputs x_i and x_j, noise_level is the value for the white noise when $x_i = x_j$ and 0 otherwise, C is the value for the constant kernel, $d(x_i, x_j)$ is the Euclidean distance between the two inputs, and l is the length scale of the kernel. The learnable hyperparameters were the white noise value, scaling constant value, and length scale of the squared-exponential kernel. Even with an optimized kernel, however, the GP will always yield some uncertainty so long as there are data gaps in the training space. As mentioned above, this uncertainty will vary depending on the distance of the training data to the inputs.

To summarize, the output of the GP is the mean of all possible functions which represent the training data. The uncertainty is input-based, with inputs "closer" to those in the training data returning lower output uncertainties than those which are "farther." The function accepts the ejecta parameters as multi-dimensional inputs and maps to a light curve output evaluated at fixed time points. Thus, each light curve is represented by a single value depicting its uncertainty. The application of this in the simulation placement scheme is discussed in Section 5.

4.2.2 Random Forest

Random forests are a subset of ensemble learning which utilize many independent decision trees to create an average group decision. Each individual tree effectively uses sequential if-else statements which separate the features of the data into categories until no further simplifications can be made. The if-else statements differ from tree to tree, resulting in potentially different outputs for each tree. After each tree has produced an output, the forest "votes" on the most common output which is taken as the interpolator output. This technique allows for the individual trees to make a decision without biased input from other trees' outputs while simultaneously averaging the outputs to exclude minority decisions [49].

Figure 7 shows a straight-forward example of some possible decisions made by trees classifying images of a cat, turtle, and bird. Both trees are supplied with the same training set, but they may not necessarily use the same decisions to separate the data into individual classes. During the classification process, the trees came up with individual classifications without any kind of cross-talk between one another. As an additional possibility, consider a third tree which may have used decisions leading to incorrect classification, such as "does it have a tail?" leading to a categorization of "yes" for the cat and bird, whose tails are visible in the training images, and "no" for the turtle, which does not have a visible tail in the training image. When presented with a test image of a turtle with a tail, the third tree may then miscategorize that image as a cat or bird. The benefit of the random forest is that all the trees' outputs are

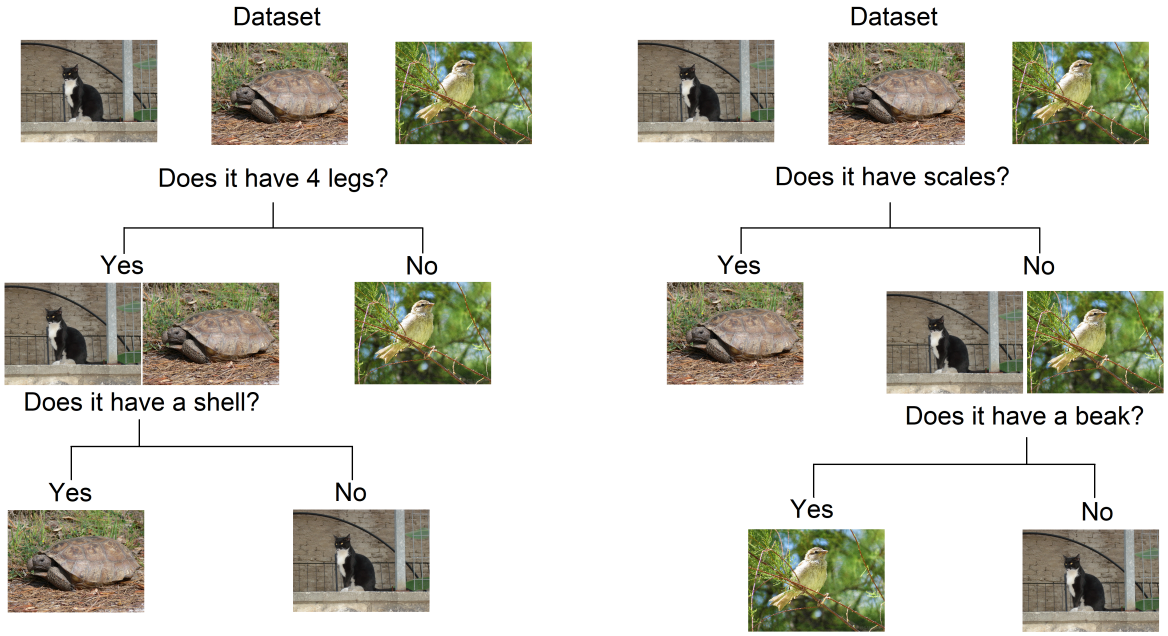

Figure 7: Two sample decision trees in the random forest for the classification of a cat, turtle, and bird. Each tree makes two decisions which allow it to separate the dataset into different categories. While the decision statements are different between the trees, given an input of a picture of a bird, both trees would be able to use their respective decisions to declare that any future input of a bird is properly classified.

averaged and only the most common is returned, so the net output would favor the correct classification by the two trees in Figure 7.

A similar decision process occurs when using random forests for regression applications. Instead of posing decision questions which are based on features, the trees attempt to sort the data by observing the training target values given the training input values. One simple example might be the relationship between the volume of water in a pot and the time required for it to boil. Given a spread of values for both variables, the trees may start off by halving the dataset in the first decision. Then, with each subsequent decision, the subsets would be further halved, until either the maximum number of branches has been reached or there is only one data point per branch, whichever is more constraining.

For this work, the standard scikit-learn decision tree class was used. The random forests also used bootstrapping, described in 4.2.3 below, but are fundamentally different from the bagging regressor in that they only select a number of features in the subset of data to analyze prior to making an estimate whereas the bagging regressor considers all the features. No error output was recorded for the random forest as it, along with the bagging regressor, served as a validation check for the Gaussian process prediction.

4.2.3 Bagging Regressor

Bagging regressors work in a similar fashion to random forests in that they are an ensemble learning tool which votes on the average output of many individual outputs. The term "bagging" is actually a portmanteau of "bootstrap" and "aggregating" [50]. Bootstrapping refers to the re-sampling of training data with replacement in order to create smaller subsets which still share the parent set's distribution. Aggregating in this sense represents the majority vote taken across the outputs of all the re-sampled subsets, similar to voting across all the outputs generated by trees in a random forest (or averaging in the case of regression). Bagging describes the manner in which training data is handled, but there is no requirement for which estimator is used to generate outputs of individual subsets. Since the scikit-learn bagging regressor class uses a decision tree by default, the same methodology as in Figure 7 holds, modulo the dataset being different for any given decision tree due to bagging. Again, it is important to reiterate that random forests only select a number of features in the subset of data to analyze prior to making an estimate whereas the bagging regressor considers all the features. In this work, both interpolators implemented bootstrapping which re-samples the data with replacement to create smaller subsets of the training set while maintaining the parent set's distribution. The bagging regressor, like the random forest, did not report an error on the interpolation as its purpose was to verify the fidelity of the Gaussian process interpolation prediction.

4.2.4 Neural Network

Two key differences set apart the previously discussed interpolators and the neural network, the latter of which will be discussed after introducing the neural network. The first is that the volume of data in the training set was not large enough to properly benefit from neural network (NN) interpolation. To alleviate this, time was added as a fifth training parameter (in addition to the ejecta masses and velocities) in the NN case. By adding ~ 200 data points per event to the initial training set of 36 events, the network could train on roughly 7000 points and create a much more valid prediction. This distinction serves to introduce context for the different approaches in Figure 10. Section 7.2 discusses the approach of including additional training parameters in more detail.

Feedforward neural networks, as the name implies, are collections of neurons in sequential layers which identify features in the data in order to create a predictive model. The neurons act as learnable weights which scale the data and subsequently feed it to an activation function. The role of the activation function is to normalize the output between layers to ensure stability and reduce the risk of vanishing or exploding gradients. The activation function of the final layer is problem-dependent, but is typically taken to be a linear function for regression applications.

More specifically, in the feedforward neural network presented in this work, each layer's output is produced from the previous layer via the layer's response $F_i : \mathbf{R}^{S_{i-1}} \rightarrow \mathbf{R}^{S_i}$, where S_i are the number of individual neurons in the network at the ith layer.

Each component of $F_i(\mathbf{p})$ (the output of each layer in the network) is characterized by an activation function

$$\mathbf{y}_i = f_i(\mathbf{w}_i\mathbf{p} + \mathbf{b}_i) \tag{18}$$

where f_i is the activation function used in the ith hidden layer of the network and $\mathbf{w}_i$ and $\mathbf{b}_i$ are the weight matrix and bias vector of the respective hidden layer. The weights and biases are learnable parameters which adapt over the course of training to better approximate the network's prediction via reference to the simulation output values for the given input parameters.

The activation function f_i chosen for all but the last layer of the network is the scaled exponential linear unit (SELU) function

$$\text{selu(x)} = \lambda \begin{cases} x & x > 0 \\ \alpha e^x - \alpha & x \leq 0 \end{cases} \tag{19}$$

with $\lambda \approx 1.0507$ and $\alpha \approx 1.6733$ as suggested in [51]. Following standard convention for regression applications, the output layer activation function is linear.

To optimize this network, the difference between the training data $\log(L_{\text{bol}})_j$ and the network's prediction is minimized. To minimize the impact of values far away from the peak luminosity, a modification to the chi-square test statistic diminishes the impact of dimmer values, while otherwise requiring agreement with the training data. This modification manifests as the second term in

$$\frac{1}{n-d}\sum_{j=1}^{n}\frac{(y_j - \log(L_{\text{bol}})_j)^2}{\sigma_j^2} \times e^{-0.2|y_j - \log(L_{\text{bol}})_{\max}|}, \tag{20}$$

where n is the number of samples, d is the dimension of the input vector (in this case the number of input physical parameters), y_j is the light curve prediction at some given time, $\log(L_{\text{bol}})_j$ is the light curve simulation output at the same given time, σ_j is the error on the simulation output, and $t_{\max}$ is the peak simulation luminosity value. Note that in Section 4.1, it was mentioned that the simulation outputs were assumed to have no error, yet an uncertainty on the training data is introduced in Equation 20. Due to the nature of the function, an error is required for evaluation; for simplicity, the error was taken to be a constant value of 10^{-3} which conveyed the notion that the simulations were assumed to be accurate while not being so small as to lead to exploding gradients during the error minimization process.

Continuing from the discussion at the start of the section, the second difference between the the Gaussian process and the neural network is in how the uncertainty is calculated by each. With the implementation of dropout in the neural network, a user-specified percentage of inputs from one layer to the next is randomly zeroed. Effectively, this means that with each pass through the neural network, the contribution of some random fraction of the training data is negated. By repeatedly sampling from the neural

network post-training with dropout enabled, the predictions are no longer deterministic but rather yield a probability distribution whose standard deviation conveys the uncertainty. For the NN, random sampling of 10000 samples with a 3% dropout resulted in an effective uncertainty measurement while the GP naturally returns the standard deviation of all the functions which could represent the training data. In this way, the NN offers an uncertainty value which is useful, but less straightforward to obtain and explain compared to the GP uncertainty.

4.3 Interpolation Discussion

Interpolation on the initial grid proved difficult due to the lack of diverse parameter combinations. Initially, to test the interpolator capabilities, a leave-one-out approach was followed where one event was removed from the initial grid and kept as a test event. The interpolators were trained on the remaining events. After training, the interpolators created light curve predictions based on the test event's parameters and the simulation output of the test event was compared to the prediction. The interpolations performed particularly poorly for events where the velocities significantly impacted the profile of the curves. Since there were only two values per velocity parameter, the interpolators learned the shape of the other light curve which was not represented by the test event. Simply put, the interpolators were being forced to extrapolate all velocity values and all but one mass value (the 0.01 $M_\odot$ mass value was between 0.001 $M_\odot$ and 0.1 $M_\odot$ and was thus the only value to not be extrapolated).

5 Automated Placement

In order to maximally explore the parameter space and reduce the amount of redundant information obtained from each new simulation, I pursued an active learning simulation placement approach. The initial collection of light curves generated using grid-spaced parameters was used as the training set for the interpolators. Thousands of parameter combinations were subsequently drawn from uniform distributions with maxima and minima matching those of the grid search in Table 2. Each of these parameter combinations was evaluated by the interpolators to produce a light curve prediction as well as an error on the whole light curve output. Eventually, the parameter combination producing the maximum error across all the tested possibilities was selected for the next placed simulation. By doing this iteratively, it is ensured that each time a simulation is placed and added to the training set, the least understood part of parameter space has some information added to it.

Currently, the interpolations evaluated to identify the largest error-producing parameters are performed on fixed time points which results in an entire light curve output evaluated at specific times. This is beneficial for the active learning process as the volume of data is minimized allowing for computational efficiency, error maximization is simplified by producing a single error value per light curve, and as seen in Figure 10,

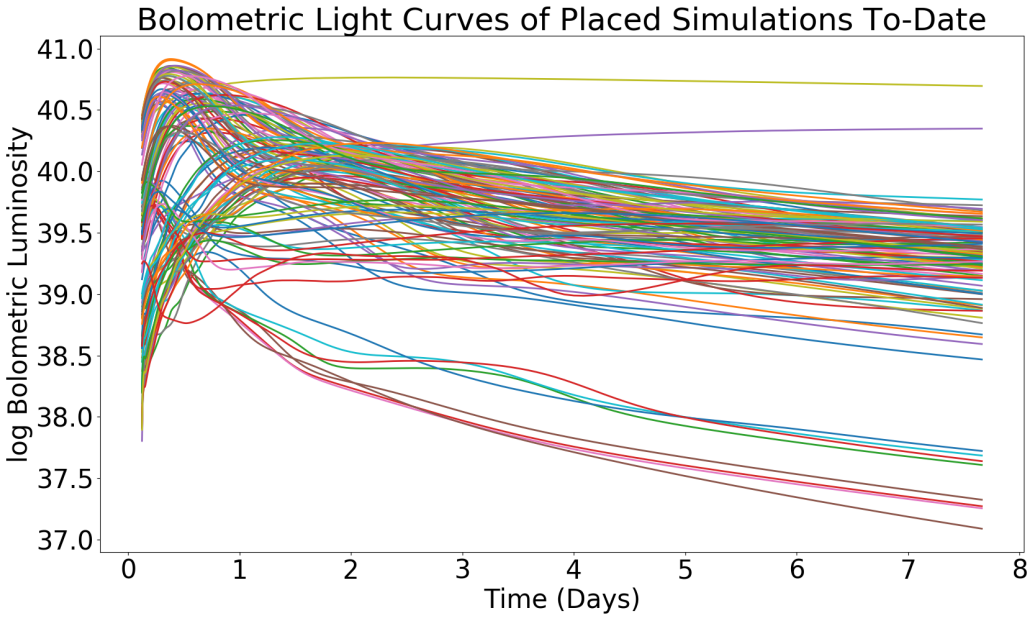

Figure 8: Bolometric light curves of the placed simulations to-date based on the Gaussian process prediction error maximization. The spaces between the previously defined upper- and mid-luminosity light curves have been thoroughly populated, but the low-luminosity grouping is still not well represented. Further discussion regarding the simulation placement behavior and statistics can be found in Section 5.4.

the fixed time interpolations produce similar enough results to the light curves which use time as a training parameter. Although the current simulation placement system is extremely efficient at placing simulations in the specified subset of the simulation population, the addition of angle as a training parameter could affect which simulations are chosen for placement (see Section 7.1 for in-depth discussion).

5.1 Initial Results

Figure 9 shows the interpolator predictions for a set of parameters identified to yield the largest Gaussian process interpolation error. The Gaussian process, unable to constrain which distribution of functions fits the training data, presents a mean value with constant uncertainty as in the prior of Figure 6. The other interpolators, including the neural network, perform well despite the poor prediction of the Gaussian process. Figure 9 is an excellent indicator of the neural network being a strong candidate for replacing the Gaussian process for simulation placement purposes.

The first 10 placed simulations were added to the original set of 36 grid-based simulations and evaluated by the interpolators to verify their impact on the improvement of the entire training set. One sample plot, Figure 10, is discussed in the main

text with the remaining plots presented in Appendix B. **It is crucial to note that in these results, each simulation being interpolated was included in the training set. The purpose of these plots was not to gauge how well unknown data was interpolated, but rather to verify that the inclusion of new placed simulations would allow interpolation of parameter combinations which, prior to placement, were not successfully interpolable by the Gaussian process**. Each event was considered in three different ways: i) the original three interpolators using only ejecta masses and velocities as input parameters, ii) the original three interpolators using time in addition to the ejecta parameters as inputs, and iii) the neural network using time in addition to the ejecta parameters as inputs.

Initially, these plots were created to verify that the inclusion of time as a training parameter would not drastically alter the interpolation outputs from those where time was not considered during training. The timing performance of the original three interpolators severely decreased when time was added during training (see Section 5.3). The neural network was considered as an interpolation option which would alleviate this timing performance issue due to its increased predictive accuracy as data volume grows. When the outputs of the original three, original three with time, and the neural network interpolators were confirmed to be generally equal, the only remaining task was to verify the uncertainties between the GP and NN. The curves and uncertainty bands in Figure 10 were made by drawing 10000 samples of the NN prediction using a dropout percentage of 3%. This particular dropout value was chosen because it effectively removed one event's collection of samples from the training set.

Based on the consistency of the outputs in Figure 10 and similar figures in Appendix B, it is obvious that the neural network interpolator is a promising future alternative for simulation placement with a larger training set. However, the smoothing of more subtle features, like the fluctuations in Figure 10, indicates that the neural network might need modification to better mimic the outputs of the original three interpolators, both with and without time. Some potential modifications to the neural network's performance are discussed in Section 7.

5.2 Training Parameter Extension

The need for the inclusion of time as a training parameter for use of the neural network interpolator invited the possibility of using other physical parameters during training. The first logical candidates after time were angle and wavelength, as they were naturally provided as part of the simulation outputs. In both cases, all the ejecta parameters identifying the simulations in the training set were used in conjunction with a set of representative wavelength or angle values. The angles were calculated using

$$\theta = \cos^{-1}\left(1 - \frac{2i}{54}\right) \tag{21}$$

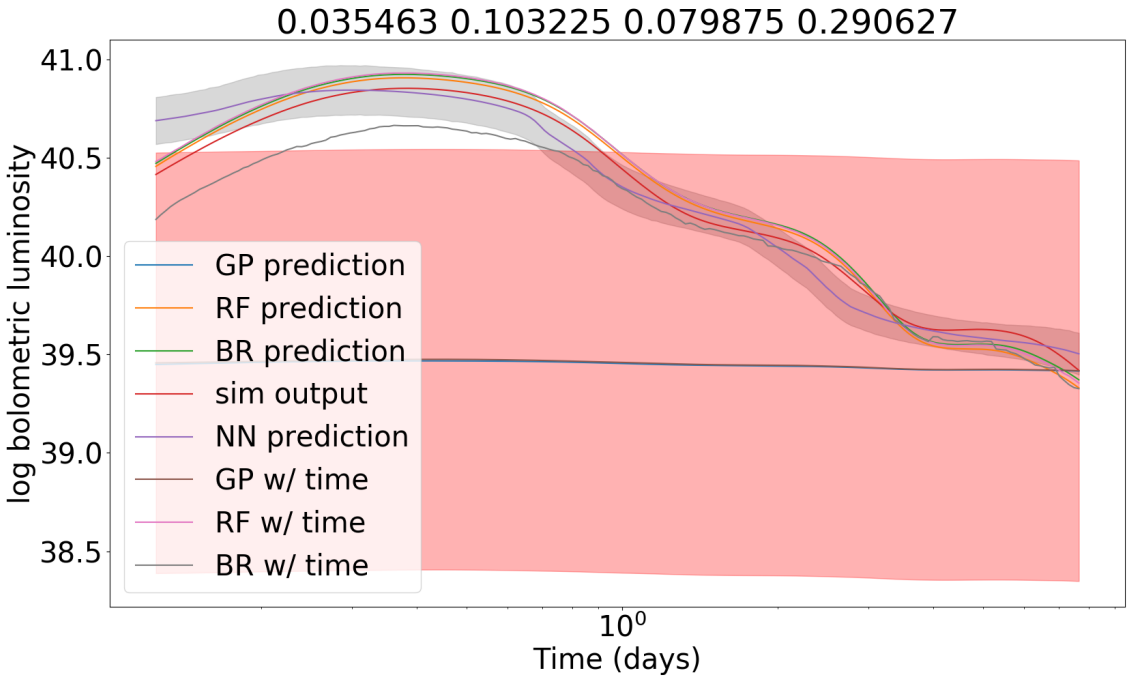

Figure 9: Example of the interpolator output prior to the placement of a large error parameter combination. The input parameters are shown in the title and correspond to dynamical mass, dynamical velocity, wind mass, and wind velocity, respectively, in units of $M_\odot$ and c. The parameters used to make this plot are the same as in Figure 10, however this is evaluation prior to the inclusion of the simulation output in the training set. The Gaussian process is unable to interpolate in this region of parameter space and, much like in Figure 6, it chooses a representative constant mean value with constant uncertainty. The other interpolators fit the data surprisingly well, giving further credence to the use of neural networks for future placement (see Sections 5.3 and 7).

where i is the i-th angular bin and all 54 bins are spaced equally in solid angle. For the purposes of constructing the training set, the angles were averaged between successive bins (i.e. averaging bins 1 and 2, bins 2 and 3, etc...). This change resulted in a 5-dimensional interpolator input of dynamical mass, dynamical velocity, wind mass, wind velocity, and angle/wavelength value. The outputs were full log bolometric luminosity light curves as used for simulation placement.

Figure 11 shows the difference between the angular bins corresponding to 0, 45, 90, 135, and 180 degree viewing angles. The differences in the peak log bolometric luminosities for the different angular bins indicate that the current approach of angle-averaging the light curves could potentially remove some non-negligible information regarding the physics of the system and the interpolation error for placement. While no predictions were made for angle as a training parameter as they currently takes days to train (see Table 3), Sections 7.1 and 7.2 further discuss the future application of angle

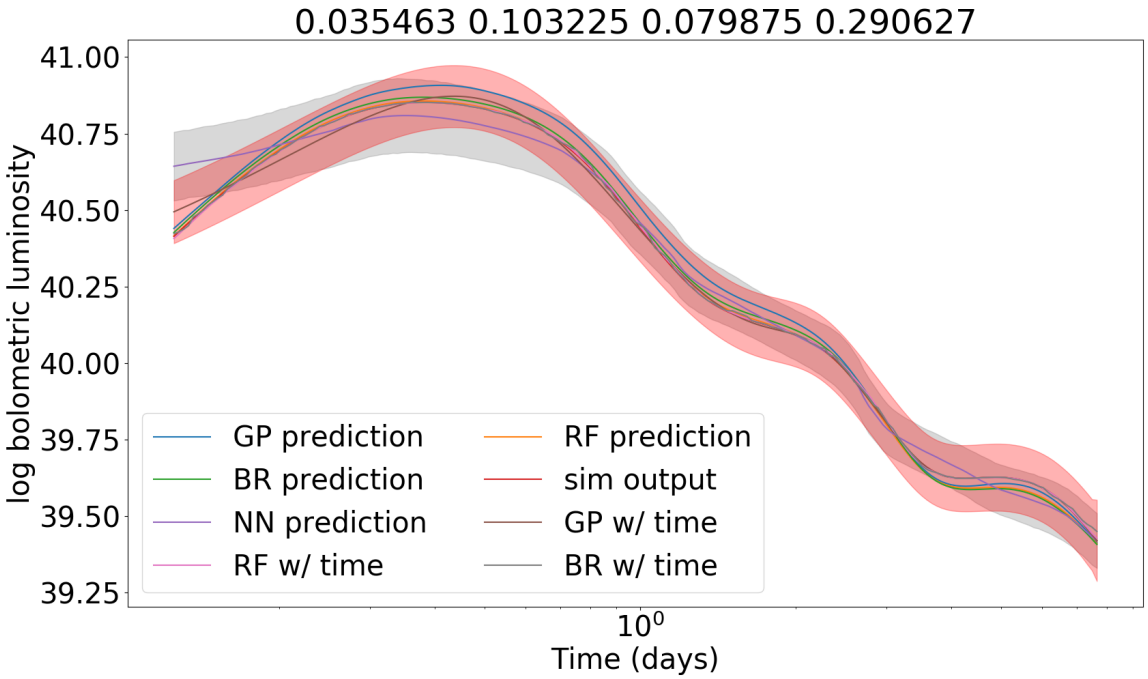

Figure 10: A comparison of the light curves generated by i) the full light curves evaluated at fixed time points which are used for simulation placement, ii) the light curves predicted by the same interpolators with the addition of time as a training parameter, and iii) the neural network, also using time as a training parameter. The red and grey bands are the 1σ uncertainty for each time point from the GP and the 1σ standard deviation of the neural network when evaluated with dropout, respectively.

as a training parameter.

A preliminary analysis of using wavelength as a training parameter is presented in Figure 12 due to its short training time. The parameter choices are the same as in Figures 10 and 9 and the outputs are being evaluated at the J-band wavelength of 1220 nm. The large dynamical range of the wavelength training parameter (476 nm for g-band to 4493 nm for Spitzer band as shown in Table 1) likely requires more simulations than were used in this particular example. This requirement explains the gap between the simulation output and the predictions within the first two days, as well as the failure to fit the late-time behavior. The jaggedness of the neural network prediction is a clear indication of insufficient data volume for that specific interpolator.

5.3 Timing Performance

The timing performance on the interpolators is incredibly efficient, with at least one interpolator training in under a minute for all cases in which only the ejecta parameters are considered during training. For the fixed time, full light curve interpolation used for

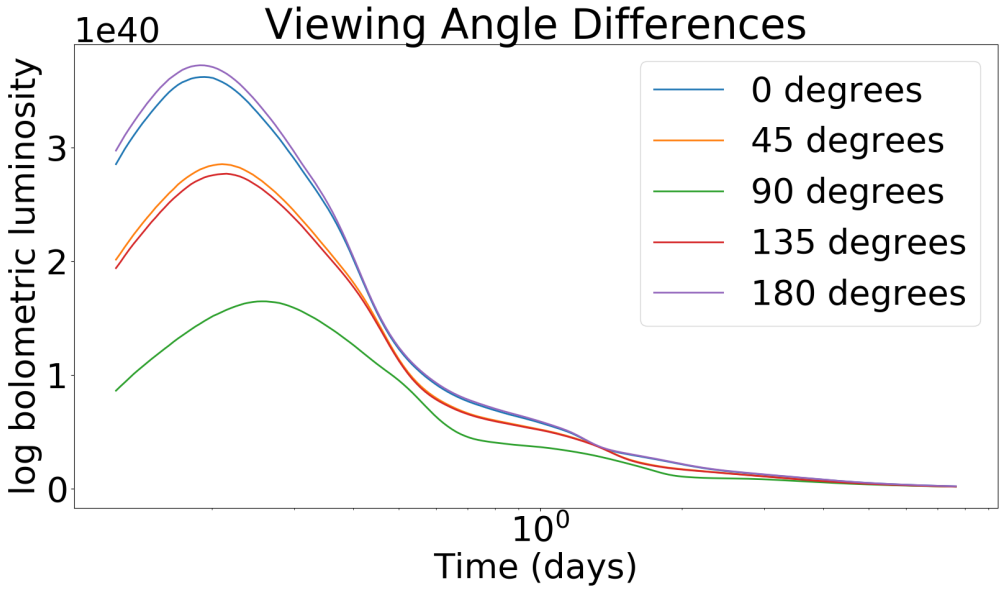

Figure 11: Light curves for different angular bins corresponding to 0^o, 45^o, 90^o, 135^o, and 180^o degree viewing angles showcasing the viewing angle dependence for a peanut-shaped morphology.

simulation placement, the original three interpolators all complete training on the order of minutes. For the case where only time is used as an additional training parameter, the neural network completes training in under a minute and predicts comparably to the multi-hour Gaussian process output as seen in Figure 10 and similar figures.

Although the inclusion of angle as a training parameter was considered for the performance metrics in Table 3, no outputs are presented as it was deemed unfeasible considering the length of the training period. As the number of placed simulations increases and the placement mechanism swaps to using the neural network prediction error, the performance metrics and outputs using angle during training will be more seriously considered.

5.4 Current Simulation Placement Status

The results presented above consider only the first 10 placed simulations which were generated near the end of May 2020. Since then, as of the writing of this thesis, roughly ~ 150 and counting simulations have been placed using the mechanism described earlier in this work. Although the training library is not yet primed for use in surrogate model generation, some initial analysis into the behavior of the placements is beneficial for evaluating long-term placement patterns.

Figure 13 shows the histogram distributions for each of the ejecta parameters in the set of placed simulations. It is immediately obvious that the most preferred

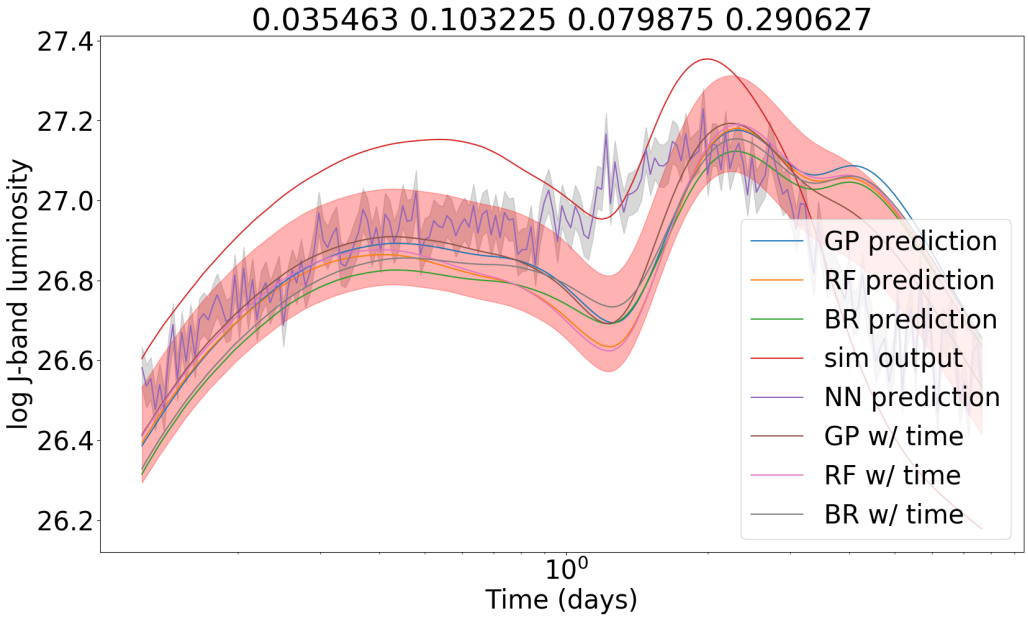

Figure 12: Prediction outputs for the same input parameters as in Figures 10 and 9 with wavelength added as an additional training variable. For this specific example, the prediction was done for the J-band wavelength at 1220 nm.

parameter values are those located near the maximum and minimum limits for each parameter ($10^{-1}M_\odot$ and $10^{-3}M_\odot$ for mass and $0.3c$ and $0.05c$ for velocity, respectively). In examining the parameter combinations for the placed simulations, it is revealed that, for most placements, only one of the four ejecta parameters is chosen to be very close to an extremum, while the other three take on less restricted values. Planned future investigation into this behavior will involve plotting the values of one parameter as a function of the others and identifying any potential correlations in parameter selection.

Another point of interest is that, over the course of placement, the interpolation error has stayed fairly constant. This constancy, in conjunction with the preference for at least one parameter near an extremum, suggests that the placement algorithm is skirting the parameter space in order to establish concrete interpolation bounds. As the purpose of the interpolators is to make predictions within the limits of the training set, the largest error is identified as anything where extrapolation is presently occurring. A similar technique of prioritizing events near the limits of the parameter space for the purpose of surrogate model generation is exhibited in Figure 2 of [52]. Presently, the belief is that after a few hundred placed simulations, the interpolators will have "closed off" the relevant parameter space and begin populating areas within the bounds, at which point the interpolation error will begin dropping. Once the error dips below a threshold found suitable for astrophysically useful predictions, the training library will be ready for surrogate model interpolation.

Training Parameters	Interpolators	Training Time
Ejecta only	GP, RF, BR	Minutes
Ejecta + Time	GP, RF, BR	Hours
Ejecta + Time	NN	Minutes
Ejecta + Wavelength	GP, RF, BR, NN	Minutes
Ejecta + Angle	GP, RF, BR	Hours/Days
Ejecta + Angle	NN	Minutes

Table 3: Rough performance metrics for the average training time of the original three interpolators and the neural network with different additional training parameters in addition to the ejecta.

6 Conclusion

With the long-term goal of astrophysically accurate surrogate model interpolation in mind, it was obvious that choosing parameters for placed simulations in a grid-like manner was not feasible. The grid-based simulations were clustered into groups as in Figures 4 and 5 which indicated that some simulations were offering repetitive information regarding sampling of the parameter space. The implementation of randomly sampled parameter combinations based on interpolation error maximization allowed for a mathematically justified mechanism of populating the less understood regions of phase space.

Perhaps most importantly, the Gaussian process error which served as the metric for new placed simulations was compared to that of the neural network with 3% dropout. The two were found to be comparable which, in conjunction with the neural network's acceptable accuracy of interpolation, served as a proof of concept that the neural network could be used for placement when the training library grows in size. This will be especially important when viewing angle is considered as a training parameter at which point the Gaussian process will take too long to evaluate interpolation prediction errors.

As the simulation placement continues, an analysis of the first $\sim$ 150 placed simulations showed that at least one of the parameters in any random error-maximized parameter combination was near an extremum value. The error for the chosen parameter combinations was also staying relatively constant and not decreasing. Although initially surprising, this is believed to stem from the interpolators attempting to skirt the edges of the parameter space. Once the clear boundaries of the phase space have been identified, there will be no possibility for extrapolation anymore, and further simulation placements will result in decreasing prediction error.

Overall, the simulation placement has worked well at exploring the parameter space in a smarter way compared to parameter selection from a grid. It will be interesting to see the interpolation error evolution as the newly placed simulations box off the relevant parameter space. Even with the placement mechanism's success, many improvements and upgrades are possible, namely the shift from using Gaussian process error to

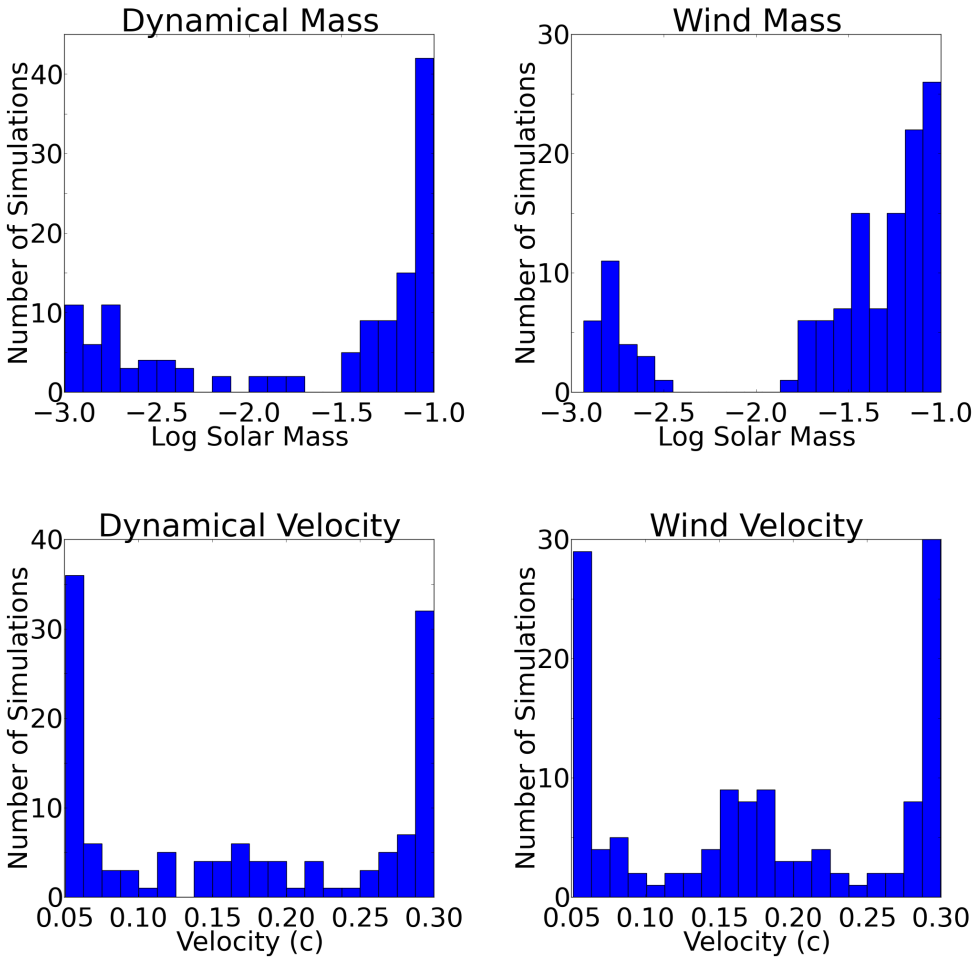

Figure 13: Histogram distributions for each of the four ejecta parameters for all the placed simulations, not including the initial grid of simulations. There is clear preference for using parameter values near the extrema.

neural network error for placement. Section 7 below discusses other potential avenues of consideration for a maximally general approach to future placement of kilonova light curve simulations.

7 Future Work

7.1 Viewing Angle as Training Parameter

Figure 11 showed the non-negligible difference in light curve behavior as a function of viewing angle. Of all the additional parameters provided by the simulation outputs which could be used as training parameters, angle is potentially one of the most significant for the long-term science return. Radiation emission will become less isotropic as

less spherically-symmetric ejecta morphologies are considered. For this reason, viewing angle will become a much more important variable as certain angular bins will convey information which others may not.

As of right now, for the sake of data consolidation, the output of each simulation is averaged in angle to reduce the number of light curves per band from 54 to 1. This averaging obviously removes information which could be pertinent to the error maximization search, especially if there is an unexpected correlation between viewing angle and interpolability. While time and wavelength were also considered as added training parameters, their contribution would be more suited for creating physics-informative models once the training set is properly populated for accurate surrogate model interpolation. With the tools currently available, it is still uncertain which interpolator would be best suited for the inclusion of angle during training; the Gaussian process takes slightly under a day to train when angle is included, but the neural network may not be supplied with enough data to reliably interpolate fixed time light curves given the training library's current state.

7.2 Interpolation in Multiple Additional Parameters

While initially created for the purpose of simulation placement through active learning, the interpolators are capable of training on more than just the four ejecta parameters as exemplified by the curves in Figure 10. Given that the simulation outputs are divided into angular bins and wavelength bands, there is clear motivation to use these as additional interpolation parameters. From the perspective of active learning, the most pressing of these would be angle so as to account for differences in viewing angle during the light curve error calculation. But, from a theoretical perspective, the inclusion of time as a training parameter is unmatched as it allows for interpolation between time points for a deeper understanding of the physics governing the light curve. And finally, from the viewpoint of an observer, the interpolation between wavelengths is essential to know what to expect when observing using a non-standard filter.

With this in mind, an attempt was made to incorporate all three of these parameters as training variables to be used solely with the neural network interpolator as the others would be unable to handle the data volume. Unfortunately, despite this and the network utilizing graphics processing units (GPUs) for increased computational power, the millions of inputs proved too overwhelming for the neural network to handle in its current state. An initial fix would be to re-attempt the training on a machine with more GPU memory and gauge the training capability there. Another workaround would involve not including all the time points in the early parts of the light curves where time sampling is denser. However, the most logical and future-proof solution would involve revisiting the neural network architecture and making it more suitable for the problem at hand. The current architecture is that of a feed-forward neural network, but it may prove more useful to consider a convolutional neural network which can learn spatial information within the light curve data as a function of angle. The potentially hours- or days-long training time required when using the entire training set of placed and initial

grid simulations is of little concern, as the interpolator handling all three parameters would be not be used for simulation placement. As such, training would be repeated at longer intervals (perhaps on the order of weeks) as more placed simulations become available and would only serve to improve the resultant surrogate models.

7.3 Morphology and Composition

As mentioned in Section 5, the wind ejecta morphology and composition were kept fixed for the simulation placement performed to date. This was initially done in order to simplify the problem while the tools were still being developed, but for maximum generality and broad consideration of all types of light curves, a new approach is necessary. Currently, the available morphologies and compositions are treated as separate classes of events. Ideally, further study of their behavior would allow for the identification of a continuous representation of these parameters. In doing so, the simulation placement tool would be generalized to accept all simulations generated within the current parameter space in addition to any new morphologies or compositions which may be considered at a later time.

In addition, the job submission tools used in the simulation placement process allow for incorporation of a user-specified composition. This capability allows for custom ejecta composition usage in the interpolations which, in tandem with previous work [30; 53], could allow for the exploration of light curve dependence on ejecta elemental composition. Given efficient and computationally inexpensive surrogate model generation, it may be possible to constrain not just the physical parameters, but also the elemental compositions of the different ejecta components.

7.4 Optimization

The implementation of the active learning process used for placement of simulations is effective in its current state. With each iteration, new parameter combinations are evaluated and the ones with the largest error are chosen to be used in the addition of a new simulation to the training library. However, many components of the interpolation and error maximization processes are unoptimized at this point.

The most important optimization issue is that the error maximization scheme is too simple in that it allows for potential overlap of placed simulations. Although the single scalar error value output by the Gaussian process is convenient, it is likely too simple a representation of the placed simulation's contribution to the training library. Currently, any given iteration can produce a nearly duplicate parameter combination which can be submitted despite a very similar simulation existing in the training set (assuming, of course, that this duplicate combination of parameters has the largest error of all tested parameters). This issue is excusable with simulation placement using SuperNu due to the available resources allowing a new simulation to be placed and completed within a matter of hours or days. However, if this approach were to be applied to more computationally expensive simulations which may take on the order

of months to complete, a much more complex error maximization technique would be essential.

It is also worth mentioning that the hyperparameters of the interpolators have not been optimized as they have been able to interpolate the entire training set on the order of minutes. While this unoptimized configuration is feasible for the current implementation of simulation placement, the performance quickly deteriorates as additional training parameters are included.

Preliminary tests of adding angle as a training parameter create a training set which requires the Gaussian process nearly a day to train while simultaneously not containing enough data for optimal neural network prediction. This may be resolved by investigating the Gaussian process parameters, namely the structure of the kernel, which could yield faster training time.

The optimal number of trees in a random forest has also not been studied, but rather set at a large enough number which yielded accurate results. Likewise, the same holds for the bagging regressor, in addition to the fact that only the decision tree estimator was used in the bagging regressor without consideration for other possibilities, such as support vector machines, k-nearest neighbors, or some other similar algorithm.

A deeper study of the neural network's hyperparameters could also allow the network to train with a smaller training library, although making the network too narrow by reducing the number of neurons per layer may result in loss of prediction accuracy.

Acknowledgments

I would like to thank my advisor, Dr. Richard O'Shaughnessy, whose invaluable mentorship and capable guidance allowed for the timely completion of this thesis. I would also like to acknowledge Dr. Saul Rappaport who helped unlock my curiosity about the physics governing the Universe and Dr. Bryce Croll for introducing me to the wonderful world of astrophysical research. Finally, I would like to thank committee members Dr. Joshua Faber and Dr. Yosef Zlochower for their contributions to the betterment of this thesis.

A Light Curve Features and Groupings

During the initial analysis of the initial grid of light curves, some notable features were identified which contributed to the groupings identified in Section 3.3. Below is a quick description of how the physical parameters influence the profiles of the light curves. The key detail to remember is that the scaling relations in Section 2.1 are for a spherically-symmetric shell of ejecta, whereas the morphologies considered in this work are peanut-shaped. As such, order-of-magnitude proportionalities will hold for the relationship between peak time and mass or velocity, but accurate predictions of peak time are more complicated due to the different morphology and two-component model structure. As mentioned before, a detailed discussion on the peak time for non-spherical morphologies is presented in [31]. The other effect of the two-component model, made apparent in the figures below, is that the wind ejecta component has a more significant impact on the shape of the light curves than the dynamical ejecta component.

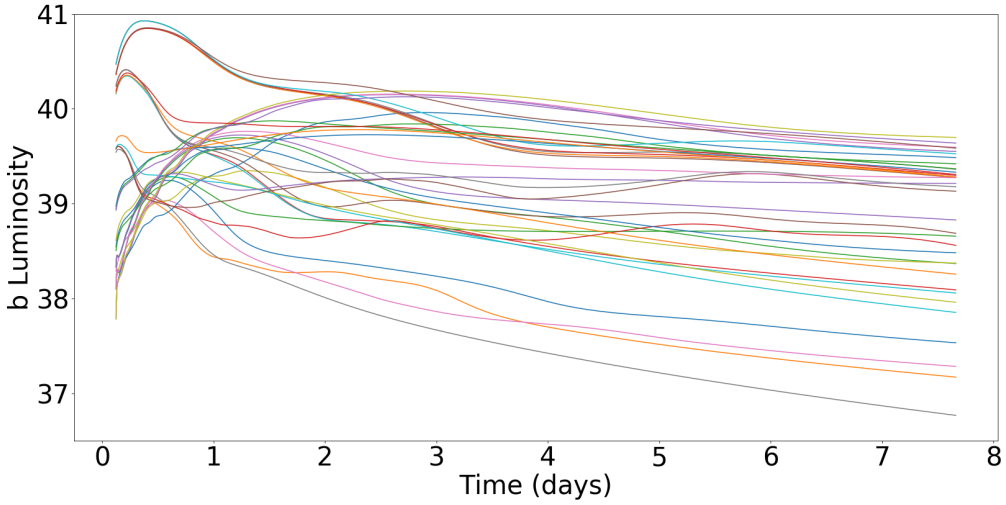

Figure 14: Initial grid light curves with no parameter constraints as in Figure 4. The y-axis is still log bolometric luminosity, here shortened to b luminosity.

A.1 Wind Velocity

Starting with a plot of all the initial grid bolometric light curves as in Figure 4 except with a linear time axis, Figure 14 shows the curve groupings are evident as before. To better differentiate between the mix of curves, a restriction is made only on the wind velocity values. From Figure 15, it is apparent that the wind velocities dictate whether the light curves rise and diminish over a long timescale or peak sharply at higher luminosity within the first day. Recalling Equation 6 and the fact that $t_{\text{peak}} \propto v^{-\frac{1}{2}}$, it

is expected that the faster-moving ejecta at the top of Figure 15 peaks sooner than the slower-moving ejecta in the bottom plot.

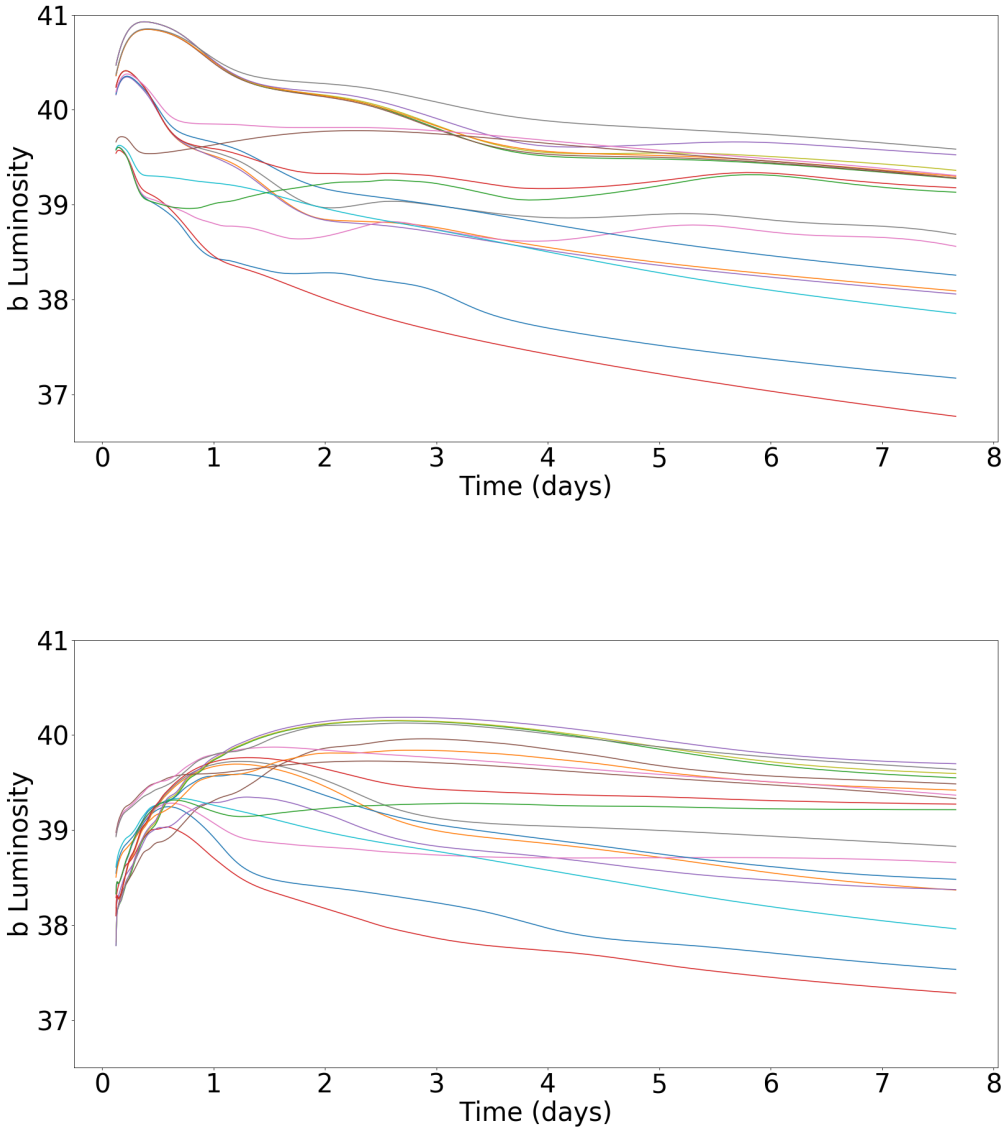

Figure 15: Initial grid light curves with the constraint that wind velocity = $0.3c$ for the top plot and wind velocity = $0.05c$ for the bottom plot.

A.2 Wind Mass

In order to identify the effects of wind mass on the light curve behavior, the top plot of Figure 15 is established as the new starting point. Limits on the wind mass, starting from the most massive at 10^{-1} $M_\odot$ and ending at the lightest 10^{-3} $M_\odot$ indicate that the wind mass most directly affects the early-time luminosity peak of the light curves as seen in Figure 16. The plots are arranged such that wind mass = 10^{-1} $M_\odot$ is at the top, wind mass = 10^{-2} $M_\odot$ is in the middle, and wind mass = 10^{-3} $M_\odot$ is at the bottom. The behavior again follows that of Equation 6, this time with $t_{\text{peak}} \propto M^{\frac{1}{2}}$, such that the higher mass ejecta peaks later than the lower-mass counterparts.

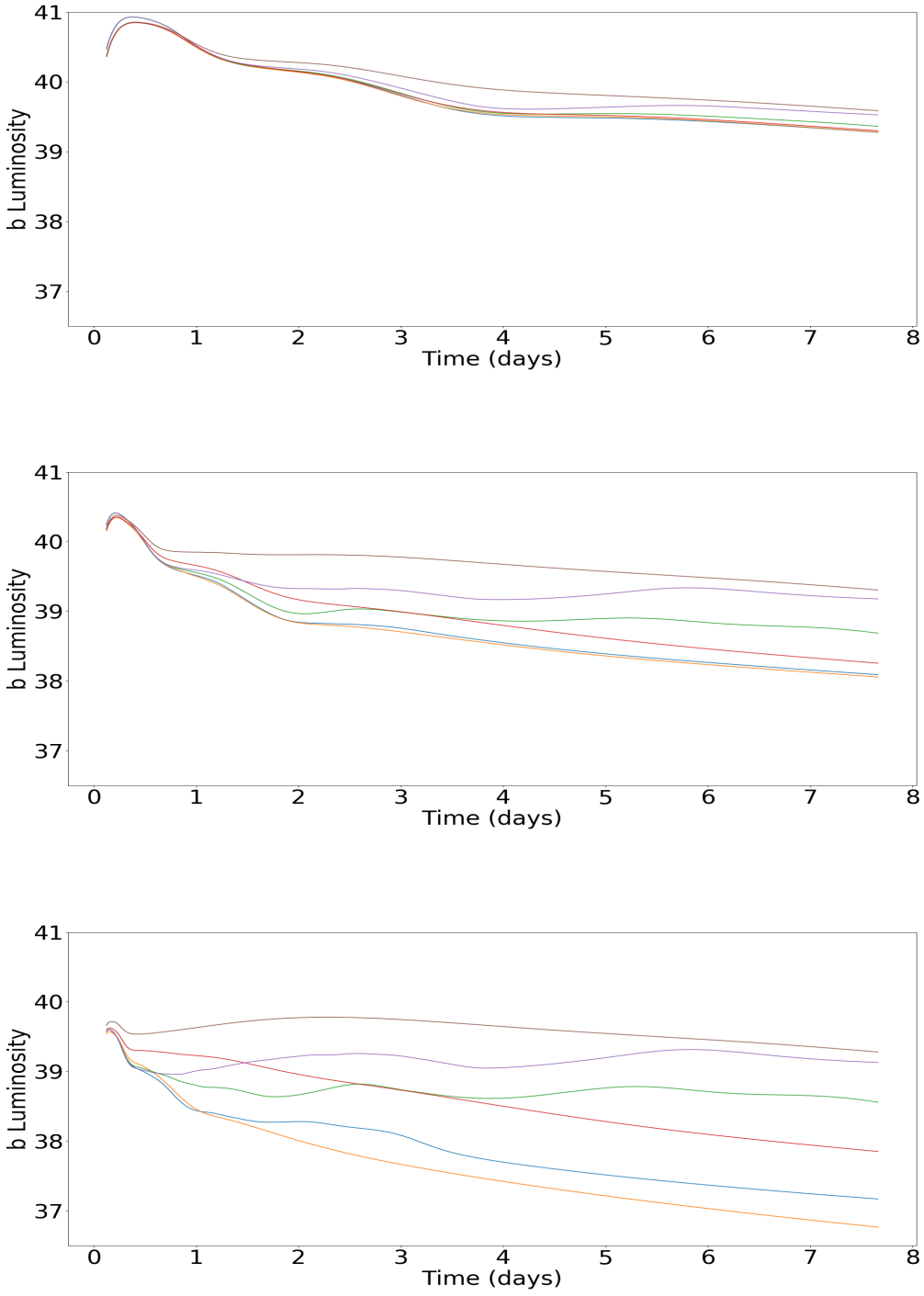

Figure 16: Initial grid light curves with the constraint that wind velocity = $0.3c$ and wind mass = 0.1 $M_\odot$ for the top plot, wind mass = 0.01 $M_\odot$ for the middle plot, and wind mass = 0.001 $M_\odot$ for the bottom plot.

A.3 Dynamical Mass

Further analysis involving constraining the dynamical mass in the bottom plot of Figure 16 shows that the dynamical mass dictates the luminosity of the light curves at later times, with higher dynamical mass resulting in brighter emission and vice versa such as in Figure 17. Again, the higher mass light curves have a much later peak time, although it is much less apparent in the dynamical mass case compared to the wind mass plots above. The remaining two light curves are naturally dictated by the remaining unconstrained variable which is dynamical velocity.

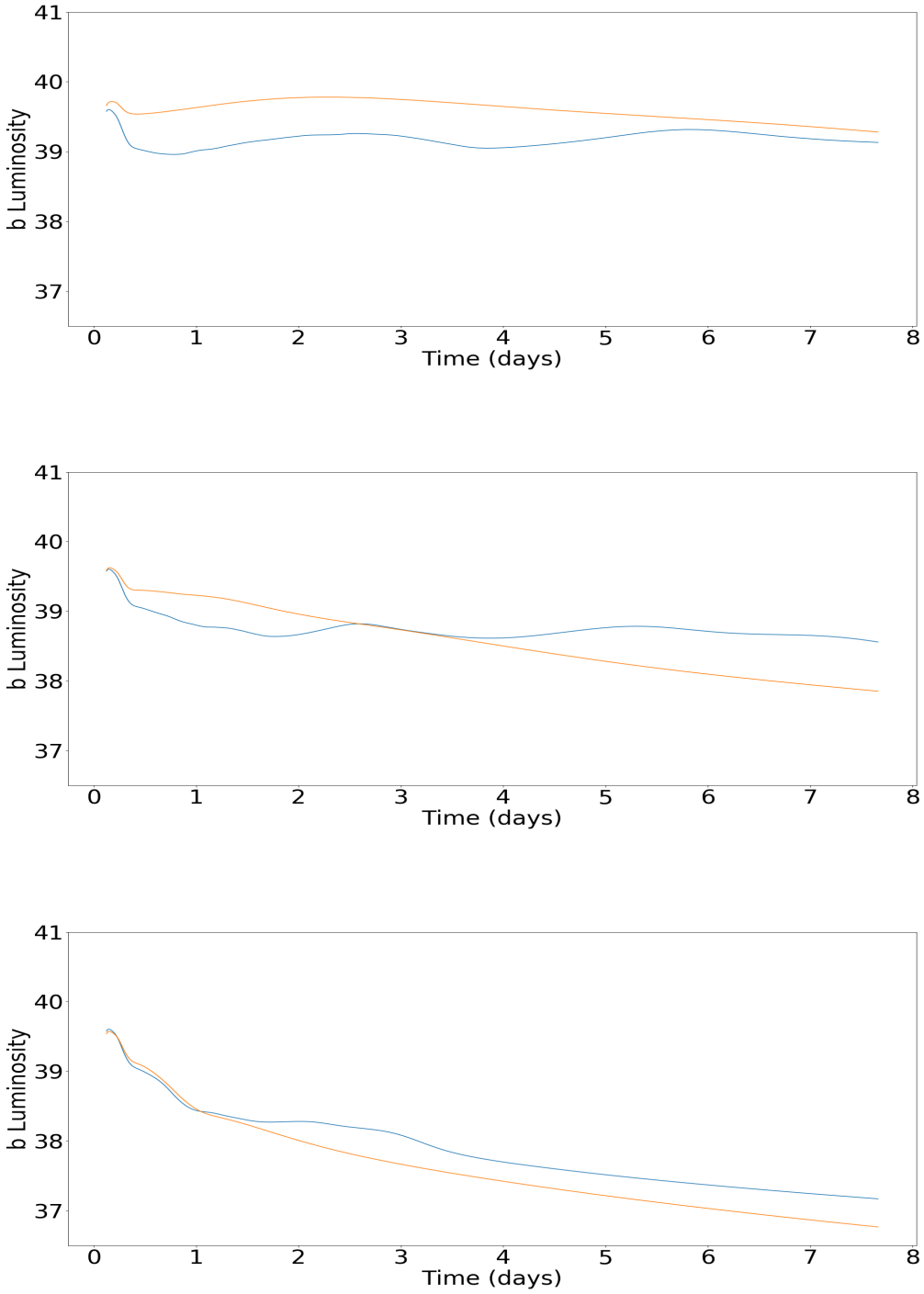

Figure 17: Initial grid light curves with the constraint that wind velocity = 0.3c, wind mass = 0.1 $M_\odot$, and dynamical mass = 0.1 $M_\odot$ for the top plot, dynamical mass = 0.01 $M_\odot$ for the middle plot, and dynamical mass = 0.001 $M_\odot$ for the bottom plot.

B Additional Initial Results Plots

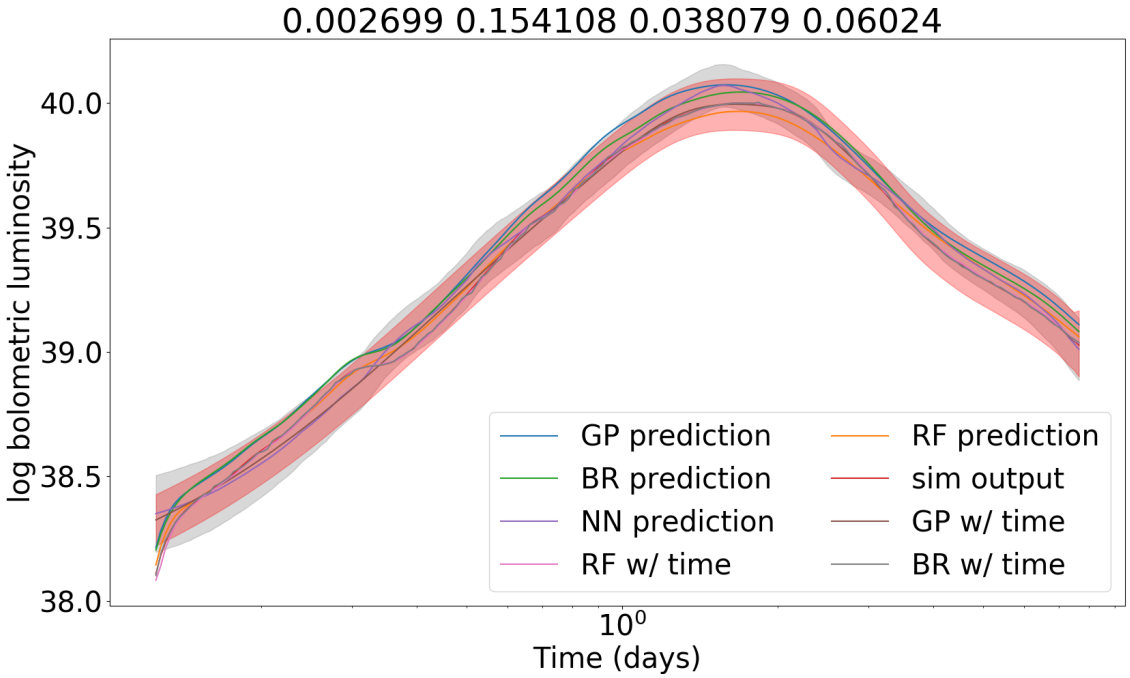

Figure 18: Same concept as Figure 10, except with different parameter combinations. Reminder that the parameters are in the title and represent the dynamical mass, dynamical velocity, wind mass, and wind velocity, respectively, in units of $M_\odot$ and c.

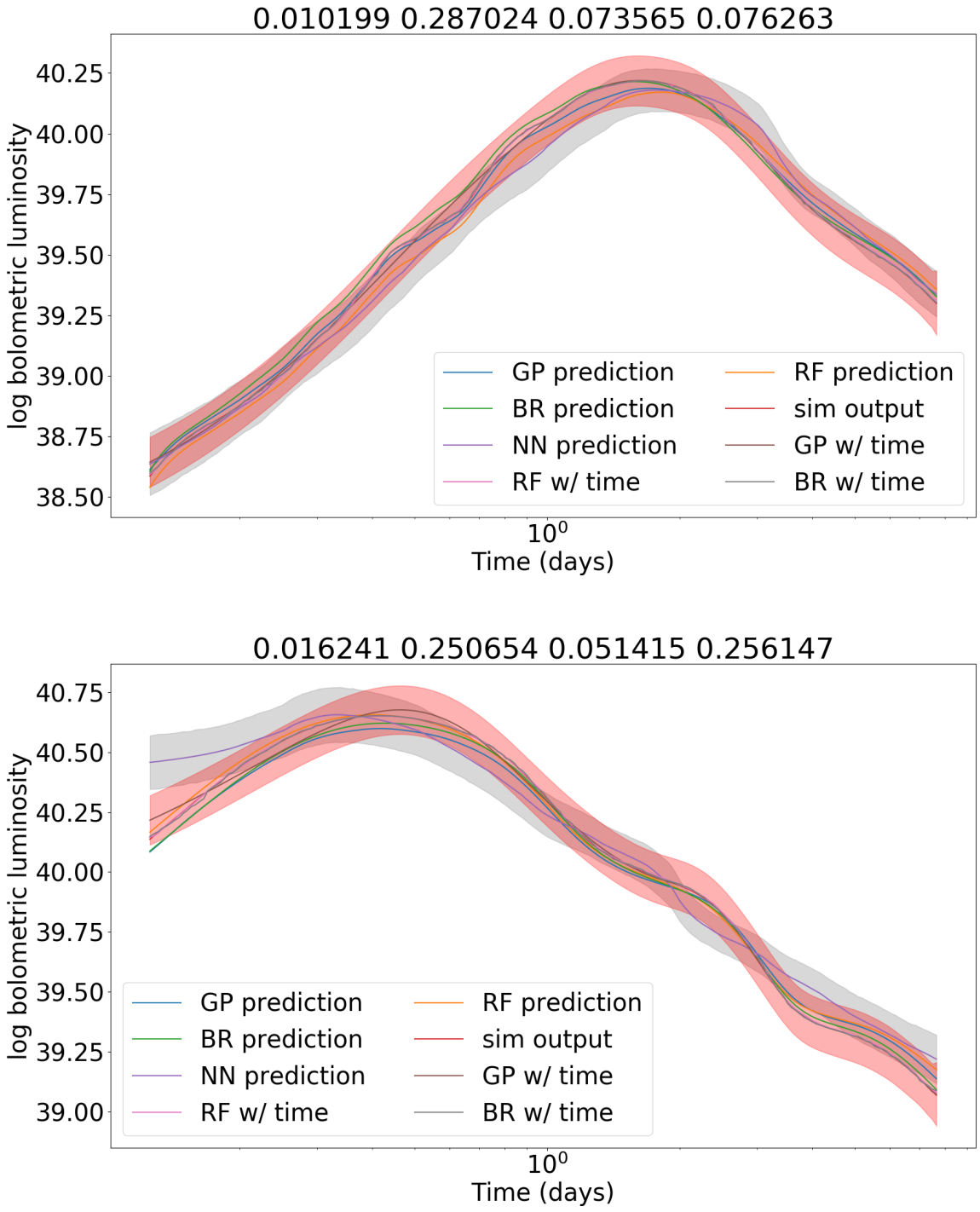

Figure 19: Same concept as Figure 10, except with different parameter combinations. Reminder that the parameters are in the title and represent the dynamical mass, dynamical velocity, wind mass, and wind velocity, respectively, in units of $M_\odot$ and c.

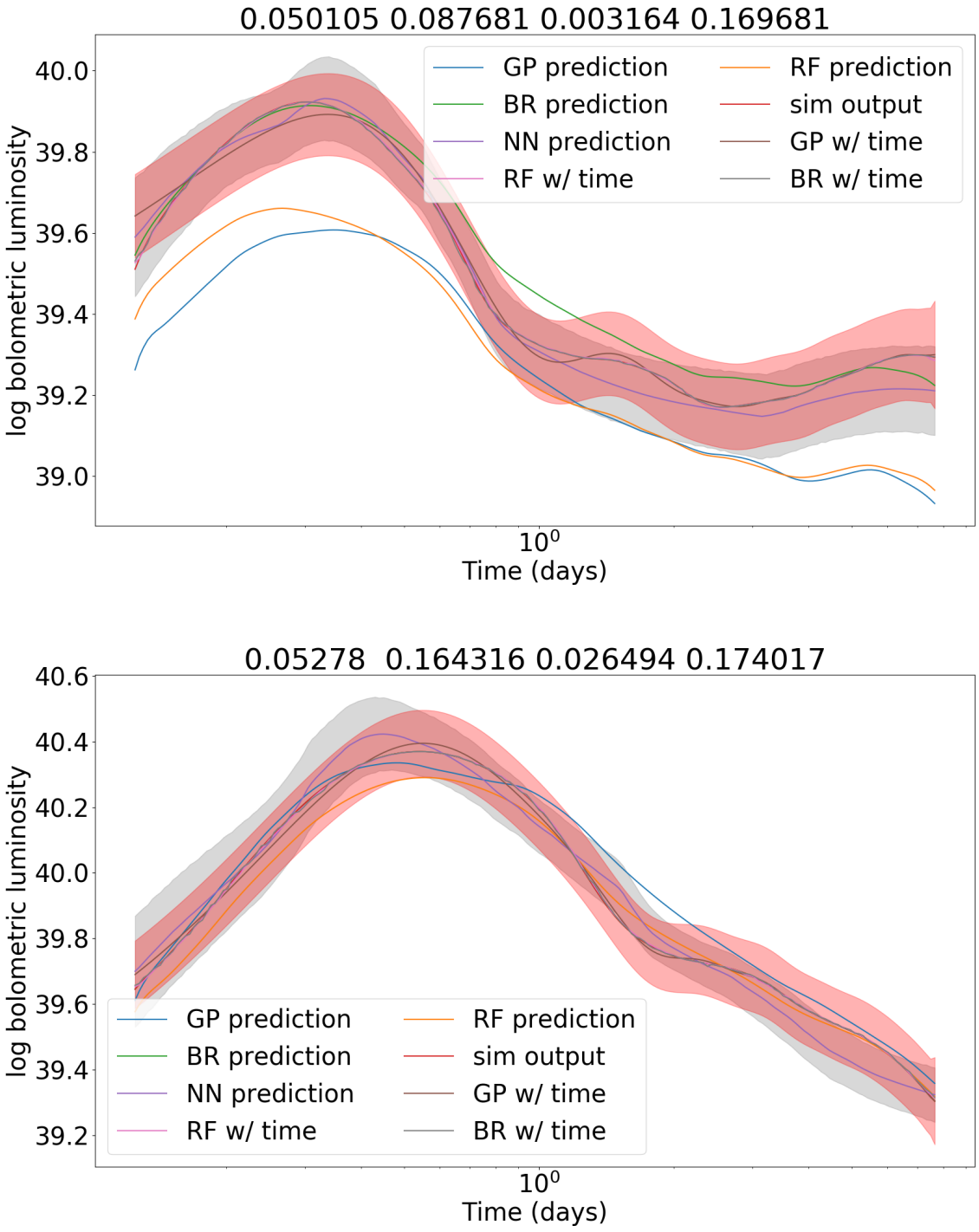

Figure 20: Same concept as Figure 10, except with different parameter combinations. Reminder that the parameters are in the title and represent the dynamical mass, dynamical velocity, wind mass, and wind velocity, respectively, in units of $M_\odot$ and c.

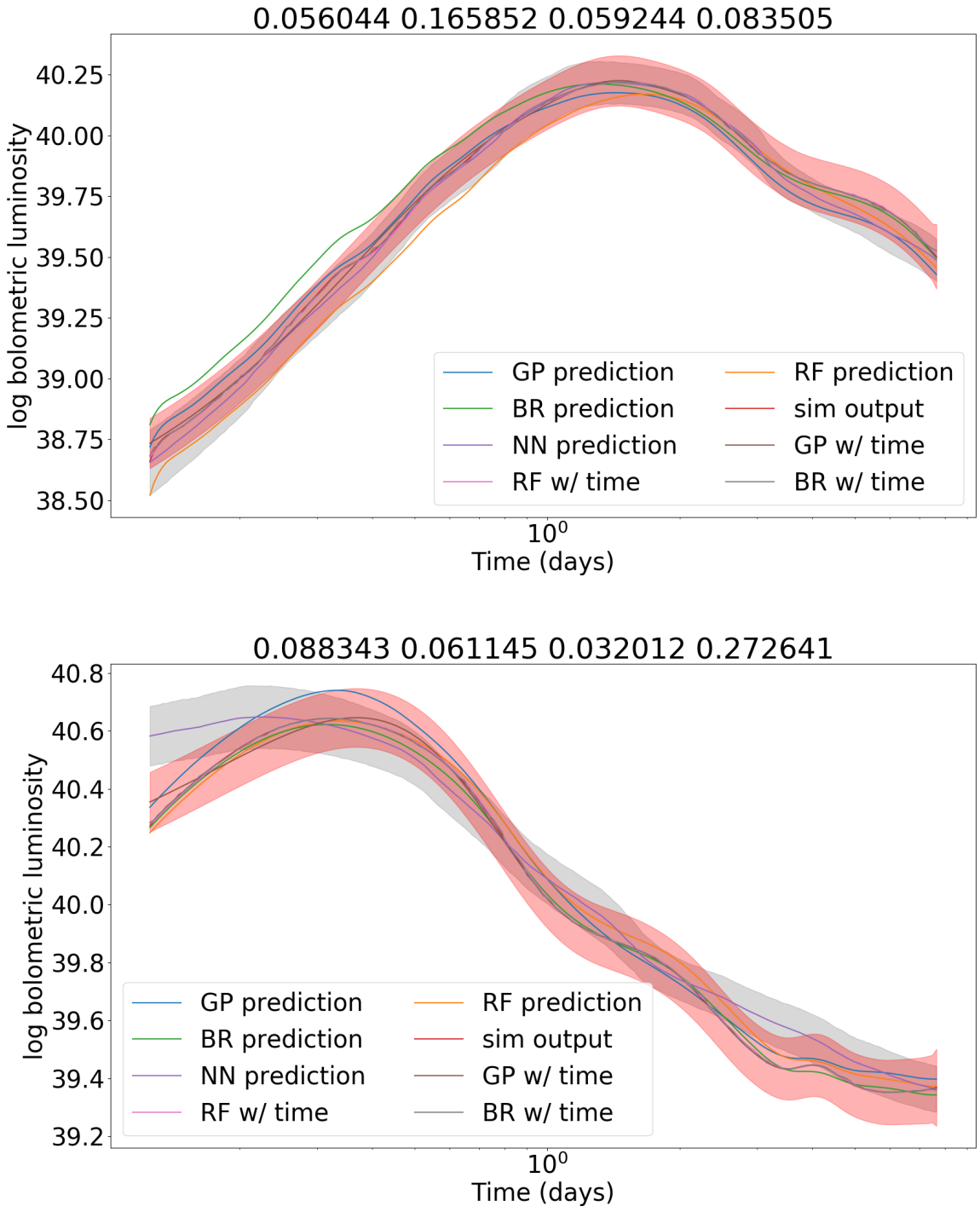

Figure 21: Same concept as Figure 10, except with different parameter combinations. Reminder that the parameters are in the title and represent the dynamical mass, dynamical velocity, wind mass, and wind velocity, respectively, in units of $M_\odot$ and c.

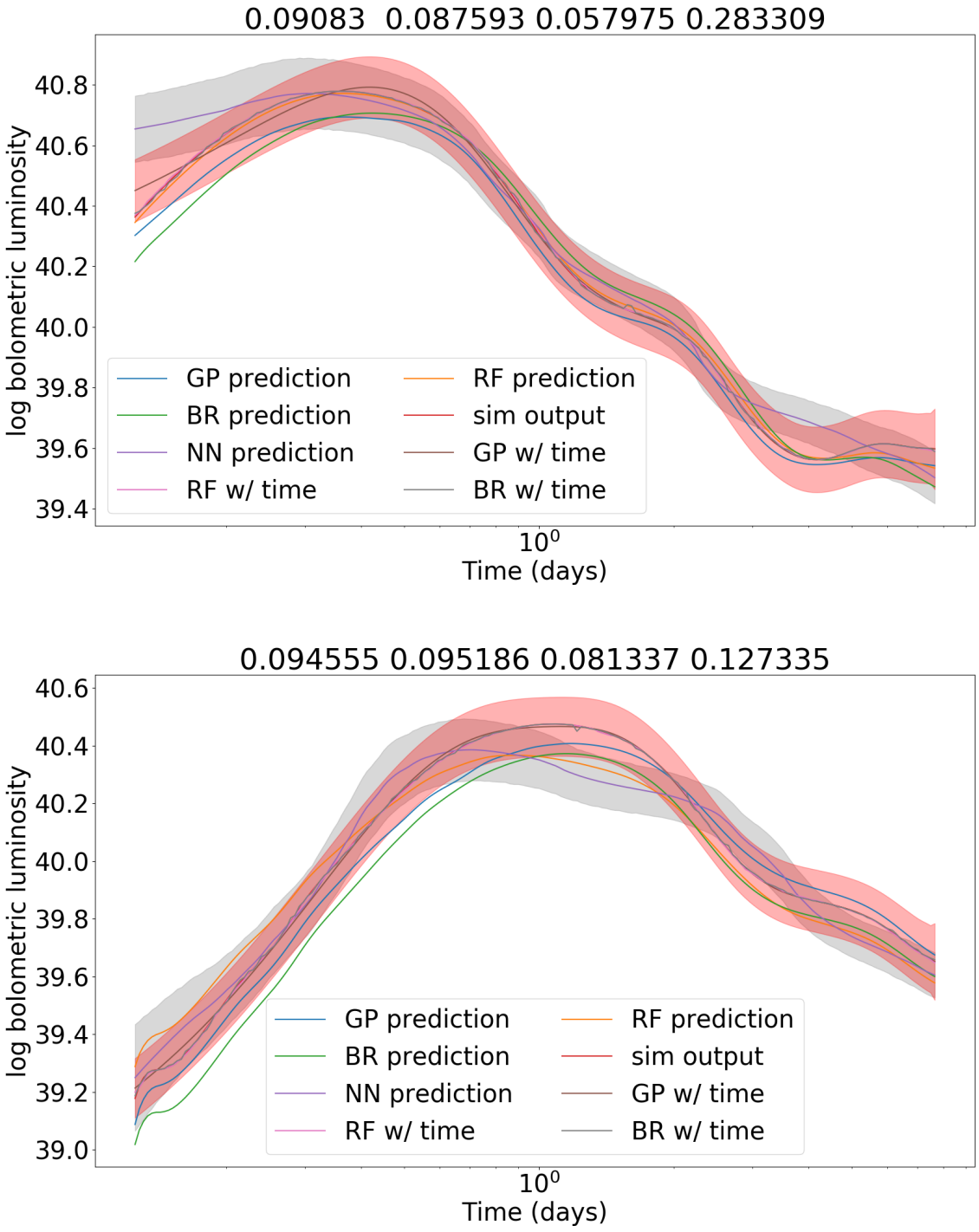

Figure 22: Same concept as Figure 10, except with different parameter combinations. Reminder that the parameters are in the title and represent the dynamical mass, dynamical velocity, wind mass, and wind velocity, respectively, in units of $M_\odot$ and c.

References

[1] J. Gunn and J. Ostriker, On the Nature of Pulsars. III. Analysis of Observations. *The Astrophysical Journal* **160** 979, 05 1970.

[2] D. Eichler, M. Livio, T. Piran, and D. N. Schramm, Nucleosynthesis, neutrino bursts and γ-rays from coalescing neutron stars. *Nature* **340**(6229) 126, July 1989.

[3] B. Abbott, et al., GW170817: Measurements of Neutron Star Radii and Equation of State. *Physical Review Letters* **121**(16), Oct 2018, ISSN 1079-7114.

[4] T. Hinderer, B. D. Lackey, R. N. Lang, and J. S. Read, Tidal deformability of neutron stars with realistic equations of state and their gravitational wave signatures in binary inspiral. *Physical Review D* **81**(12), Jun 2010, ISSN 1550-2368.

[5] R. A. Hulse and J. H. Taylor, Discovery of a pulsar in a binary system. *ApJL* **195** L51, January 1975.

[6] B. P. Abbott, et al., Observation of Gravitational Waves from a Binary Black Hole Merger. *Phys. Rev. Lett.* **116** 061102, Feb 2016.

[7] B. P. Abbott, et al., GWTC-1: A Gravitational-Wave Transient Catalog of Compact Binary Mergers Observed by LIGO and Virgo during the First and Second Observing Runs. *Phys. Rev. X* **9** 031040, Sep 2019.

[8] L.-X. Li and B. Paczyński, Transient Events from Neutron Star Mergers. *ApJL* **507** L59, November 1998.

[9] The LIGO Scientific Collaboration, et al., Multi-messenger Observations of a Binary Neutron Star Merger. *ApJL* October 2017.

[10] M. Soares-Santos, et al., The Electromagnetic Counterpart of the Binary Neutron Star Merger LIGO/Virgo GW170817. I. Discovery of the Optical Counterpart Using the Dark Energy Camera. *ApJL* **848** L16, October 2017.

[11] M. Shibata, et al., Modeling GW170817 based on numerical relativity and its implications. *Phys. Rev. D* **96** 123012, Dec 2017.

[12] D. M. Siegel, GW170817 –the first observed neutron star merger and its kilonova: Implications for the astrophysical site of the r-process. *The European Physical Journal A* **55**(11), Nov 2019, ISSN 1434-601X.

[13] The LIGO Scientific Collaboration, et al., Estimating the Contribution of Dynamical Ejecta in the Kilonova Associated with GW170817. *ApJL* **850** L39, December 2017.

[14] M. Ruiz, S. L. Shapiro, and A. Tsokaros, GW170817, general relativistic magnetohydrodynamic simulations, and the neutron star maximum mass. **97**(2) 021501, January 2018.

[15] A gravitational-wave standard siren measurement of the Hubble constant. *Nature* **551**(7678) 8588, Oct 2017, ISSN 1476-4687.

[16] D. Kasen, B. Metzger, J. Barnes, E. Quataert, and E. Ramirez-Ruiz, Origin of the heavy elements in binary neutron-star mergers from a gravitational-wave event. *Nature* **551**(7678) 8084, Oct 2017, ISSN 1476-4687.

[17] M. R. Drout, et al., Light curves of the neutron star merger GW170817/SSS17a: Implications for r-process nucleosynthesis. *Science* **358** 1570, December 2017.

[18] D. Radice, A. Perego, F. Zappa, and S. Bernuzzi, GW170817: Joint Constraint on the Neutron Star Equation of State from Multimessenger Observations. *ApJL* **852** L29, January 2018.

[19] M. W. Coughlin, et al., Constraints on the neutron star equation of state from AT2017gfo using radiative transfer simulations. *MNRAS* **480** 3871, November 2018.

[20] J. A. Faber and F. A. Rasio, Binary Neutron Star Mergers. *Living Reviews in Relativity* **15**(1), Jul 2012, ISSN 1433-8351.

[21] B. D. Metzger, Kilonovae. *Living Reviews in Relativity* **23**(1), Dec 2019, ISSN 1433-8351.

[22] A. Bauswein, S. Goriely, and H.-T. Janka, SYSTEMATICS OF DYNAMICAL MASS EJECTION, NUCLEOSYNTHESIS, AND RADIOACTIVELY POWERED ELECTROMAGNETIC SIGNALS FROM NEUTRON-STAR MERGERS. *The Astrophysical Journal* **773**(1) 78, Jul 2013, ISSN 1538-4357.

[23] J. R. Oppenheimer and G. M. Volkoff, On Massive Neutron Cores. *Phys. Rev.* **55** 374, Feb 1939.

[24] B. D. Metzger, A. L. Piro, and E. Quataert, Time-dependent models of accretion discs formed from compact object mergers. *Monthly Notices of the Royal Astronomical Society* Sep 2008, ISSN 1365-2966.

[25] B. D. Metzger, A. L. Piro, and E. Quataert, Neutron-rich freeze-out in viscously spreading accretion discs formed from compact object mergers. *Monthly Notices of the Royal Astronomical Society* **396**(1) 304314, Jun 2009, ISSN 1365-2966.

[26] D. Arnett, Analytic solutions for light curves of supernovae of Type II. *The Astrophysical Journal* **237** 541, 03 1980.

[27] D. Arnett, Type I supernovae. I - Analytic solutions for the early part of the light curve. *The Astrophysical Journal* **253** 785, 01 1982.

[28] E. Chatzopoulos, J. Craig Wheeler, and J. Vinko, GENERALIZED SEMI-ANALYTICAL MODELS OF SUPERNOVA LIGHT CURVES. *The Astrophysical Journal* **746**(2) 121, Jan 2012, ISSN 1538-4357.

[29] D. Grossman, O. Korobkin, S. Rosswog, and T. Piran, The long-term evolution of neutron star merger remnants - II. Radioactively powered transients. *MNRAS* **439** 757, March 2014.

[30] R. T. Wollaeger, et al., Impact of ejecta morphology and composition on the electromagnetic signatures of neutron star mergers. *Monthly Notices of the Royal Astronomical Society* **478**(3) 3298, 04 2018, ISSN 0035-8711.

[31] O. Korobkin, et al., Axisymmetric Radiative Transfer Models of Kilonovae. 2020.

[32] O. Korobkin, S. Rosswog, A. Arcones, and C. Winteler, On the astrophysical robustness of the neutron star merger r-process. *Monthly Notices of the Royal Astronomical Society* **426**(3) 19401949, Oct 2012, ISSN 0035-8711.

[33] B. W. Carroll and D. A. Ostlie, An Introduction to Modern Astrophysics (2007), 2nd (international) edition.

[34] K. Hotokezaka, S. Wanajo, M. Tanaka, A. Bamba, Y. Terada, and T. Piran, Radioactive decay products in neutron star merger ejecta: heating efficiency and -ray emission. *Monthly Notices of the Royal Astronomical Society* **459**(1) 35, 04 2016, ISSN 0035-8711.

[35] B. D. Metzger and E. Berger, WHAT IS THE MOST PROMISING ELECTROMAGNETIC COUNTERPART OF A NEUTRON STAR BINARY MERGER? *The Astrophysical Journal* **746**(1) 48, Jan 2012, ISSN 1538-4357.

[36] L. S. Collaboration, et al., LSST Science Book, Version 2.0. 2009.

[37] M. Cohen, W. A. Wheaton, and S. T. Megeath, Spectral Irradiance Calibration in the Infrared. XIV. The Absolute Calibration of 2MASS. *The Astronomical Journal* **126**(2) 1090, aug 2003.

[38] G. G. Fazio, et al., The Infrared Array Camera (IRAC) for the Spitzer Space Telescope. *The Astrophysical Journal Supplement Series* **154**(1) 10, sep 2004.

[39] K. Kawaguchi, M. Shibata, and M. Tanaka, Diversity of Kilonova Light Curves. *The Astrophysical Journal* **889**(2) 171, feb 2020.

[40] C. J. Fontes, et al., Relativistic opacities for astrophysical applications. *High Energy Density Physics* **16**, 6 2015.

[41] C. Fontes, C. Fryer, A. Hungerford, R. Wollaeger, and O. Korobkin, A line-binned treatment of opacities for the spectra and light curves from neutron star mergers. *Mon. Not. Roy. Astron. Soc.* **493**(3) 4143, 2020.

[42] R. T. Wollaeger and D. R. van Rossum, RADIATION TRANSPORT FOR EXPLOSIVE OUTFLOWS: OPACITY REGROUPING. *The Astrophysical Journal Supplement Series* **214**(2) 28, oct 2014.

[43] K. D. Alexander, et al., The Electromagnetic Counterpart of the Binary Neutron Star Merger LIGO/Virgo GW170817. VI. Radio Constraints on a Relativistic Jet and Predictions for Late-time Emission from the Kilonova Ejecta. *ApJL* **848** L21, October 2017.

[44] E. Waxman, E. O. Ofek, D. Kushnir, and A. Gal-Yam, Constraints on the ejecta of the GW170817 neutron star merger from its electromagnetic emission. *Monthly Notices of the Royal Astronomical Society* **481**(3) 34233441, Sep 2018, ISSN 1365-2966.

[45] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu, A Limited Memory Algorithm for Bound Constrained Optimization. *SIAM Journal on Scientific Computing* **16**(5) 1190, 1995.

[46] F. Pedregosa, et al., Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* **12** 2825, 2011.

[47] A. Paszke, et al., PyTorch: An Imperative Style, High-Performance Deep Learning Library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett, editors, Advances in Neural Information Processing Systems 32, pages 8024–8035 (Curran Associates, Inc., 2019).

[48] C. Rasmussen and C. Williams, Gaussian Processes for Machine Learning. Adaptive Computation and Machine Learning (MIT Press, Cambridge, MA, USA, 2006).

[49] L. Breiman, Machine Learning, Volume 45, Number 1 - SpringerLink. *Machine Learning* **45** 5, 10 2001.

[50] L. Breiman, Bagging Predictors. *Machine Learning* **24** 123, 1996.

[51] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter, Self-Normalizing Neural Networks. In I. Guyon, et al., editors, Advances in Neural Information Processing Systems 30, pages 971–980 (Curran Associates, Inc., 2017).

[52] V. Varma, S. E. Field, M. A. Scheel, J. Blackman, L. E. Kidder, and H. P. Pfeiffer, Surrogate model of hybridized numerical relativity binary black hole waveforms. *Physical Review D* **99**(6), Mar 2019, ISSN 2470-0029.

[53] W. Even, et al., Composition Effects on Kilonova Spectra and Light Curves: I. 2019.